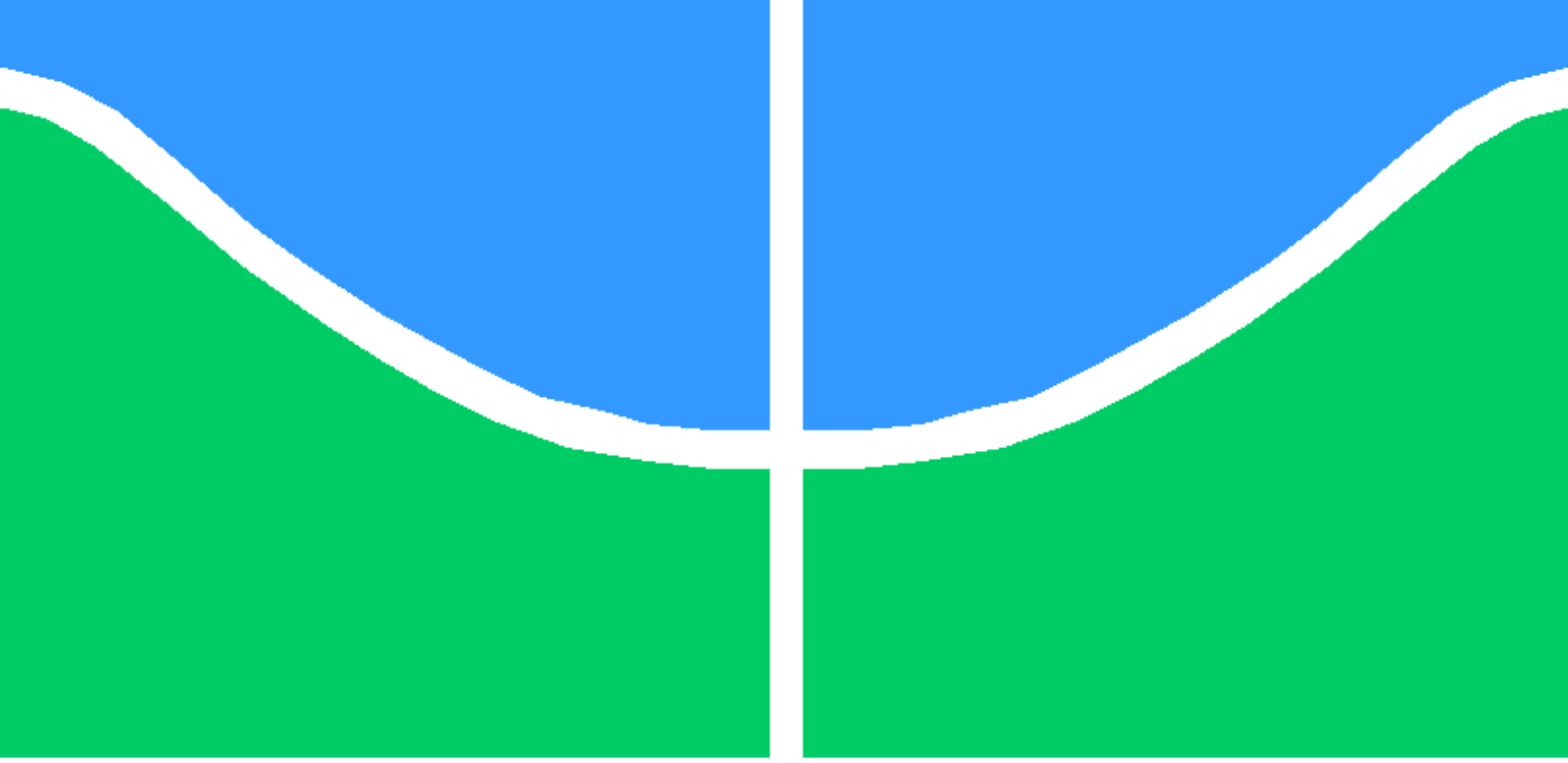

PROJETO FINAL DE GRADUAÇÃO 2

Ferramenta para validação de permissões de acesso a serviços da plataforma Android

Felipe Oliveira Araujo Pereira

Paulo Henrique Ferreira Campos Mendes

Brasília, Dezembro de 2019

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

PROJETO FINAL DE GRADUAÇÃO 2

Ferramenta para validação de permissões de acesso a serviços da plataforma Android

Felipe Oliveira Araujo Pereira

Paulo Henrique Ferreira Campos Mendes

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Engenheiro de Redes de Comunicação

Banca Examinadora

Prof. Dr. Flavio Elias Gomes de Deus, ENE/UnB _____
Orientador

Prof. Dr. Alexandre Solon Nery, ENE/UnB _____
Examinador

Prof. Dr. Georges Daniel Amvame Nze, ENE/UnB _____
Examinador

"Medicina, Direito, Administração, Engenharia, são atividades nobres, necessárias à vida. Mas a poesia, a beleza, o romance, o amor, são coisas pelas quais vale a pena viver."

Sociedade dos Poetas Mortos, 1989

“Se enxerguei mais longe, foi porque me apoiei sobre os ombros de gigantes".

Traduzido, Sir Isaac Newton

"Deus no comando."

Paulo H. F. C. Mendes

Agradecimentos

Dedico esse trabalho à minha família pelo suporte incondicional que me ofereceram ao longo dessa caminhada; a minha namorada, Fernanda, pela compreensão, carinho e apoio em todos os momentos difíceis; aos meus colegas de curso que me consolaram numa mesa de bar. Agradeço ao meu colega e coautor deste trabalho, Paulo, pela parceria ao longo de toda a graduação. Que seu sucesso seja tão grande quanto seu bom humor e perseverança.

Felipe Oliveira Araujo Pereira

Agradeço a Deus e a espiritualidade amiga pela força constante ao longo do caminho, à minha família pela paciência e o suporte que nunca faltou, aos colegas e professores por terem me ensinado todos os dias e , por fim, ao professor Flávio Elias e ao Laboratório LATITUDE/LabForense pelas portas que me foram abertas e por todas as oportunidades de crescimento como pesquisador e engenheiro.

Paulo Henrique Ferreira Campos Mendes

SUMÁRIO

LISTA DE FIGURAS

RESUMO

Dada a popularização de dispositivos móveis com o sistema operacional Android e o aumento notável de crimes cibernéticos que envolvem a aquisição e a venda de informações confidenciais por meio de aplicativos móveis, fica evidente a necessidade de desenvolver ferramentas que possam fornecer aos usuários o quão vulnerável estão as suas informações pessoais.

Este trabalho apresenta uma ferramenta desenvolvida para verificar as permissões de acesso a serviços no sistema Android, confrontando as mesmas com as permissões informadas na Google PlayStore e, por fim, informar o quão transparente é a informação oferecida pela loja de aplicativos e, simultaneamente, possibilitar a análise de quais tipos de informação podem ser acessadas pelo aplicativo em análise.

ABSTRACT

Given the popularization of mobile devices running Android operating system and the notable increase in cyber crimes involving the acquisition and sale of sensitive information through mobile applications, there is a clear need to develop tools that can provide users with an opinion of how vulnerable their private information will be to the applications used on their mobile devices.

This paper presents a tool developed to check the access permissions to services on the Android system, comparing them with the permissions informed on Google PlayStore and, finally, inform how transparent is the information offered by the app store and simultaneously enable the analysis of what types of information can be accessed by the application under review.

Capítulo 1

Introdução

As duas primeiras décadas do século XXI trouxeram consigo a popularização dos dispositivos móveis, como *smartphones* e *tablets*, e em seguida a conexão em massa desses dispositivos com a Internet. Segundo dados de 2017(CISCO, 2017), cerca de 8.6 bilhões de pessoas estavam conectadas à Internet através de dispositivos móveis e a estimativa feita pela mesma empresa é que esse número chegue a 12.3 bilhões de pessoas em 2022.

Um dos impactos mais notáveis causados pela massificação do acesso mundial à Internet é a mudança em diversos modelos de negócios até então estabelecidos como os serviços de transporte, educação, marketing, bancos e comércio de uma forma geral. Essas mudanças também trouxeram uma crescente preocupação com a privacidade e prevenção contra acesso indevido de terceiros a informações confidenciais do usuário.

Assim como a quantidade de usuários, os ataques a *smartphones* também se tornaram numerosos e isso principalmente pelo fato desses dispositivos armazenarem informações que podem ser facilmente monetizadas como senhas, dados de cartões de crédito, dados de localização e informações institucionais. As fraudes ocorridas nos últimos anos, como o desenvolvimento de aplicativos *fakes* disponibilizados nas lojas (DURAN, 2018) e aplicativos legítimos que venderam dados de localização (GUARDIAN, 2018), mostram o impacto causado pelo roubo de dados privados e a necessidade do desenvolvimento de estratégias que combatam os crimes cibernéticos e protejam a privacidade do usuário.

O sistema operacional Android, em sua versão 9.0 (*Pie*) conta com a gritante quantia de 324 permissões possíveis (DEVELOPERS, 2019d) de serem concedidas a aplicativos que as solicitarem, e, de acordo com a sua classificação de sensibilidade em permissões comuns, permissões de assinatura e permissões sensíveis (DEVELOPERS, 2019j), e como contramedida aos casos de ataques supracitados, conta com a opção do usuário de conceder ou não acesso aos aplicativos instalados aos serviços disponíveis (contanto que estes estejam explicitados no manifesto de publicação do aplicativo na plataforma de distribuição de aplicativos para sistemas Android, PlayStore).

Antes de dar início a discussão que move este artigo, é relevante pontuar a crescente necessidade do celular como artigo básico no dia a dia moderno, e em como o usuário comum é feito refém das comodidades oferecidas pelo dispositivo.

1.1 Dispositivos móveis como instrumentos essenciais

A evolução da capacidade de processamento de dispositivos móveis e a facilidade de acesso a Internet serviram como combustível para a popularização desse tipo de dispositivo. Associado então com a infinitude de aplicações disponíveis para essa plataforma, há grande disponibilidade de um dispositivo que já foi considerado supérfluo.

Afirmar que os dispositivos móveis são símbolos da era em que vivemos é pouco mediante a importância socioeconômica dessa modalidade de aparelho. No âmbito social, temos o aparelho celular como artefato que, além de sua função utilitária, é usado para constituir a si mesmo e ao mundo. Troca de mensagens, transferência de fotos, agendamento de encontros e consultas, gravação de vídeos, gerenciamento de arquivos, pagamentos e finanças, lazer, solicitação de transporte, compartilhamento de localização, etc. Há uma infinidade de aplicações disponíveis para *download* na plataforma *GooglePlay* (serviço fornecido pela Google, onde o usuário pode procurar, localizar, visualizar e fazer *download* de conteúdo para seu dispositivo móvel). Algumas dessas aplicações são desenvolvidas pela própria empresa que fabrica o aparelho, oferecendo ao então seu conjunto de aplicações nativas. Mas, em sua grande maioria, o usuário opta por utilizar aplicações desenvolvidas por empresas terceiras. Dessa forma, colocando sua informação para acesso sob demanda dessas empresas que pouco se sabe o que fazem com estes dados.

A partir do contexto apresentado, levanta-se o questionamento: o quão sensível é a informação que se transporta em um dispositivo móvel e qual o fim dado a elas pelas empresas que administram e mantém os aplicativos?

1.2 Dispositivos móveis e o modelo de acesso a serviços por meio de permissões

Na seção 2.1.5 será explorado de maneira aprofundada o funcionamento de como o sistema operacional Android concede aos aplicativos solicitantes permissões de acesso a serviços que retornam uma ou mais variáveis, como por exemplo informações sobre a galeria de imagens do usuário, lista de contatos e telefones ou informações completas sobre a Wi-Fi a qual o dispositivo está conectado. Por ora, é de interesse desenhar o contorno dessas atribuições e segmentá-las em grupos genéricos de forma a facilitar o compreensão de suas usabilidades assim como nos é oferecido na listagem de permissões solicitadas pelos aplicativos disponibilizados na plataforma PlayStore:

- Histórico do app e do dispositivo;
- Fotos/mídia/arquivos;
- Câmera;
- ID do dispositivo e informações da chamada;
- Identidade;
- Contatos;
- Telefone;

- Armazenamento;
- Microfone;
- Local;
- SMS;
- Informações sobre a conexão Wi-Fi;
- Outros;

É de vital importância ressaltar que tais permissões compõem o principal objeto de estudo deste documento, como descrito na sessão 3.2 e que seus pormenores serão devidamente explorados a fim de validar a solução proposta.

De acordo com a listagem de permissões (DEVELOPERS, 2019d) que podem ser concedidas pelo sistema operacional Android (tendo em mente as limitações de cada aparelho que possui o SO instalado, por alguns podem não conter, por exemplo, acesso à Internet, câmera ou gerenciador de mensagens de texto SMS), tal categorização é apenas uma forma de facilitar a leitura de usuários menos familiarizados com as solicitações que dado aplicativo fará ao dispositivo.

Para justificar a existência dos serviços oferecidos pelo sistema operacional nos dispositivos móveis, é preciso evidenciar casos nos quais este melhora a vida do usuário, assim como expor situações onde esse tipo de informação do usuário tem uso duvidoso pela aplicação solicitante.

A seguir serão apresentados, apenas a título de ilustração, diversos cenários em que a informação de localização do usuário é exposta. Esse tipo de exposição pode acontecer com outros tipos de dados provenientes de outras permissões como leitura de SMS, leitura de contatos, informações de rede e gravador de áudio. Assim torna-se notável que a coleta de dados por si só não é um fim, mas provê embasamento para diversas aplicações.

1.2.1 O sistema de localização na saúde.

A proposta *GeoHealth* (HENRIQUE GONCALVES SA et al., 2012) apresentada pelos autores como projeto de pós graduação consiste em uma aplicação que utiliza desse tipo de informação do usuário um sistema para coleta, registro e análise de dados de localização na atenção primária, utilizando módulos *mobile* e *web* para acompanhamento epidemiológico na região oeste do estado de São Paulo. Hoje, o sistema *GeoHealth* está em produção e evidenciou melhoras no sistema de atendimento do SUS-SP, com um custo baixíssimo por habitante por mês, evita-se o manejo de papel, propõe melhoras na atenção primária à região e alimenta a base de dados do DATASUS em conjunto dos demais sistemas que também fazer parte do DATASUS.

Exergames (KAMEL BOULOS; YANG, 2013) são jogos eletrônicos que captam e virtualizam os movimentos reais dos usuários e compõem grande parcela dos lançamentos do gênero de realidade aumentada no mercado de *games*. Dentro dessa categoria, encaixa-se o mundialmente popular aplicativo *Pokémon Go*. O aplicativo tem como principal motor a captura da posição em tempo real do usuário, propondo um universo virtual com interação do jogador. Desde sua data de lançamento em 2016 até hoje estudos levantados por (ALTHOFF; WHITE; HORVITZ, 2016) e (GABBIADINI; SAGIOGLOU; GREITEMEYER, 2018) acerca do impacto devido ao uso contínuo

do jogo, seus reflexos nos hábitos dos usuários e contribuição ao quadro clínico de doenças físicas e distúrbios psico-sociais. No que tange a contribuição à saúde física do usuário, (GABBIADINI; SAGIOGLOU; GREITEMEYER, 2018) apresentam melhorias consideráveis na prática fomentada pelo jogo em curtos prazos de até 3 meses de uso diário, enquanto (ALTHOFF; WHITE; HORVITZ, 2016) expõe em uma janela tempo maior a parábola no comportamento saudável do usuário. O artigo escrito por (KAMEL BOULOS; YANG, 2013) aborda extensivamente o tema de *exergames* como gamificação de interações geossociais por meio de aplicações e *gadgets*.

1.2.2 O sistema de localização na mobilidade urbana

Aqui a informação de localização do usuário é usada a fim de estabelecer padrões comportamentais e a relação do indivíduo com a cidade em que habita. O estudo comportamental, pode ser segmentado em duas vertentes: a perspectiva individual onde o foco principal está na identificação de trajetórias individuais, sua regularidade e a sua relação com atributos sociais dos usuários; enquanto a perspectiva urbana empenha-se em avaliar o conjunto de interações entre deslocamentos de pessoas de uma mesma cidade, construindo assim um sistema complexo. Ainda no âmbito da perspectiva urbana, o levantamento de métricas a partir de informação provinda de dispositivos móveis ou pontas de prova estáticas na cidade mostra-se de extrema valia para a otimização do transporte.

Não é adequado considerar a mobilidade do indivíduo isoladamente, ou seja, sem considerar seu contexto social, econômico e cultural, a partir do cenário em que o usuário avaliado, é possível traçar métricas acerca de seus costumes de mobilidade: sua distribuição (de onde o usuário veio e qual o seu destino, como as viagens do usuário se distribuem no espaço físico), frequência, modalidade de transporte escolhida e caminhos utilizados. A partir dessa informação do usuário, retoma-se a discussão proposta previamente: o que é feito com essa informação?

O estudo feito por (ZAIATZ, 2018) realizado em uma amostra de 86 estudantes no ano de 2018, no cenário do Campus da Universidade de Brasília evidencia, a partir de dados coletados via *API Google Maps*, importantes aspectos da vida do estudante, como a região que este habita, hábitos de deslocamento e até mesmo um estudo social que pode ser estudado com embasamento adequado por demais áreas das ciências sociais.

Destaca-se a leitura de (CASTRO et al., 2013), onde é evidenciado por meio de pesquisa e análise da informação proveniente do usuário a importância dos táxis equipados com dispositivos que levantam *footprints* e *logs*, contribuindo então para a otimização de trajetos em dada cidade. A partir da *dinâmica operacional* adotada pelos motoristas, é possível prever comportamentos anormais no trânsito, avaliar percursos com mais demandas de clientes e, se estressada a ideia, até mesmo observar bairros evitados por estes.

Alinhado às ideias anteriores, (ROSE; LTD, 2006) levanta diferente metodologia para aquisição da informação tratada em seu texto, e foca em avaliar o impacto do dispositivo móvel como ponta de prova no trânsito, possibilitando inferência em detecção de acidentes, facilidade de administração do trânsito através de semáforos e rampas comandadas por um operador (aqui cabe avaliar que,

na época de publicação desse artigo, Internet das Coisas ainda não era um assunto em voga, visto que este tomou proporções de debate científico no início da década de 2010; atualmente há a possibilidade de dispositivos reguladores de trânsito inteligentes que inferem a partir da informação coletada), prover informação ao cidadão, facilitando assim as decisões tomadas no trânsito, como caminho a percorrer, trajetos a evitar e tempo de viagem e, por fim, levanta métricas que esboçam quantitativamente a situação do trânsito.

1.2.3 O sistema de localização como solução de segurança

O uso da ferramenta de localização também pode promover a segurança e, quando aliado a demais ferramentas (como oxímetros de pulso, compassos eletrônicos e semelhantes), garantir a integridade física do usuário. Oferecer vigilância e monitoramento de presos em cárcere, julgados em regime semi-aberto ou prisão domiciliar também é uma aplicação a qual a ferramenta demonstra seu valor. No ambiente online, por meio da informação de localização proveniente, por exemplo, em um simples *login* em sua rede social preferida, pode provar que alguém do outro lado do mundo tentou acessar sua conta.

No Brasil, a ideia de adotar o monitoramento eletrônico já era cogitada desde 2007 em São Paulo, o que foi aprovado só em 2008 em alguns estados. E apenas em 2010, foi regulamentado o monitoramento eletrônico a nível nacional, através da implementação da Lei n. 12.528/2010. O monitoramento eletrônico consiste, em regra, no uso de um dispositivo eletrônico pelo "criminoso" (não necessariamente apenas os efetivamente condenados, bastando que figurem como réus em um processo penal condenatório), que passaria a ter a liberdade (ainda que mitigada ou condicionada) controlada via satélite, evitando que se distancie ou se aproxime de locais predeterminados. Este dispositivo indica a localização exata do individuo, uma vez que o sistema permite saber, com precisão, se a área delimitada está sendo obedecida. Já com isso possibilita o registro de sua movimentação pelos operadores da central de controle. O tema é profundamente explorado por (SILVA, 2016), defendendo a legitimidade do equipamento como alternativa ao cárcere e seus impactos da vivência do condenado, facilitando sua reinserção na sociedade.

Antes de apresentar o próximo estudo, é importante ressaltar que este corresponde a um cenário avaliado em Portugal, onde a entidade reguladora CNPD (Comissão Nacional de Proteção de Dados) estabelece diretrizes quanto ao campo de ação das empresas, garantindo a integridade e privacidade do trabalhador. No Brasil, ainda não há um órgão responsável pela jurisdição estabelecida pela Lei 13.709/2018; ainda no texto da Lei, são descritas as competências da Autoridade Nacional de Proteção de Dados (ANPD) — aprovada sob categorização de Medida Provisória em 25 de Maio de 2019 — e as competências do Conselho Nacional de Proteção de Dados Pessoais e da Privacidade — colegiado será composto por 23 titulares não remunerados, com mandato de dois anos, e de diferentes setores: seis do Executivo Federal; um do Senado Federal; um da Câmara dos Deputados; um do Conselho Nacional de Justiça; um do Conselho Nacional do Ministério Público; um do Comitê Gestor da Internet no Brasil; quatro da sociedade civil com atuação comprovada em proteção de dados pessoais; quatro de instituição científica, tecnológica e de inovação; e quatro de entidade do setor empresarial ligado à área de tratamento de dados pessoais.

Apresentado por (PRUDENTE, 2014) como um aspecto cinza do uso do serviço de localização em dispositivos móveis tem-se a finalidade de vigilância sobre o trabalhador exercida pela empresa contratante. Como instrumento disciplinar, o GPS é ferramenta que garante que o trabalhador cumpra com ordens e normas da empresa, assegurando ao empregador que medidas corretivas como repreensão registrada, sanções pecuniárias, perda de dias de férias e suspensão do trabalho com abatimento em salário sejam dadas como legais por parte do empregador, como afirma o Código de Trabalho português. Porém, cabe avaliar que tal restrição e aparato exerça sobre o trabalhador o trauma psicológico de constante vigilância, interferindo em suas atividades e produtividade esperada, gerando estresse laboral.

1.3 Definição do Problema

O ano de 2018 ficou marcado por alguns crimes cibernéticos dentre os quais se destacaram: i) a monetização de dados de localização coletados ilegalmente em aplicativos gratuitos para IOS (GUARDIAN, 2018) o roubo de informações pessoais em aplicativos de rede social em sistemas Android através de um trojan descoberto pela Trustlook Labs (TRUSTLOOK, 2018), o caso Facebook & Cambridge Analytica onde dados de mais de 50 milhões de usuários foram utilizados sem a devida ciência dos mesmos (KOTTASOVÁ, 2018). Os três casos anteriormente citados, além de não formarem uma lista exaustiva, são exemplos de situações em que dados sensíveis foram utilizados de maneira indevida.

1.4 Objetivos

Nesse cenário fica evidente o quanto o usuário pode ter suas informações expostas por aplicações maliciosas que rodam em seus dispositivos móveis. Na mesma proporção em que o número de ameaças e ataques cresce, é notável a necessidade da pesquisa e do desenvolvimento de ferramentas capazes de combater potenciais situações de risco a integridade e preservação da privacidade e segurança do usuário.

O objetivo geral deste projeto é mostrar o desenvolvimento de uma ferramenta para dispositivos Android capaz de analisar as permissões de acesso à informações sensíveis. O objetivo geral foi dividido nos seguintes objetivos específicos:

- Estudar sistema Android, suas APIs, providers e a cadeia de permissões.
- Promover alterações no código-fonte do Android para tornar acessível informação acerca das permissões utilizadas;
- Monitorar o arquivo AndroidManifest.xml para leitura e contestação das permissões utilizadas com API da Google PlayStore.

Capítulo 2

Fundamentação Teórica

Este capítulo trata dos principais aspectos teóricos relacionados ao funcionamento do sistema Android, sua arquitetura em camadas, a cadeia de permissões e a função do AndroidManifest.

2.1 O Android como sistema operacional

Um sistema operacional (SO) pode ser compreendido como um conjunto de aplicações desenvolvidas na forma de softwares que são responsáveis por gerir os recursos de hardware - CPU, memória, sensores e outros - e assim permitir que as mais variadas aplicações possam fazer uso dos recursos de hardware de um dispositivo. Em síntese um SO é uma interface que faz a abstração entre a camada de hardware e as aplicações em software permitindo assim padronização na entrada e saída de dados, como pode ser visto na Figura 2.1(LEARNINGOS, 2018) .

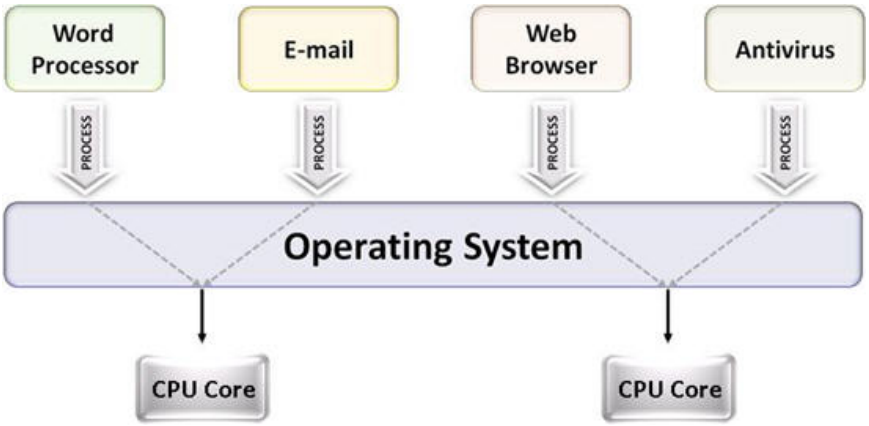

Figura 2.1: Abstração feita pelo SO entre as aplicações e a camada de hardware. (LEARNINGOS, 2018)

No caso dos dispositivos móveis há uma série de SOs populares como o IOS, o Windows Mobile, Symbian OS, Blackberry OS e muitos outros, contudo o que tem maior domínio do mercado de SOs móveis é o Android. Sua primeira versão foi desenvolvida em 2003 pela Android inc. e atualmente

é mantido pela Google, está na versão 9 (Pie) e é de código aberto. É o sistema operacional móvel mais utilizado no mundo segundo dados do site de estatísticas StatCounter (STATCOUNTER, 2019), e mesmo tendo como foco os dispositivos com telas sensíveis ao toque é utilizado também em TVs, relógios de pulso, vídeo games, câmeras digitais e até mesmo em carros.

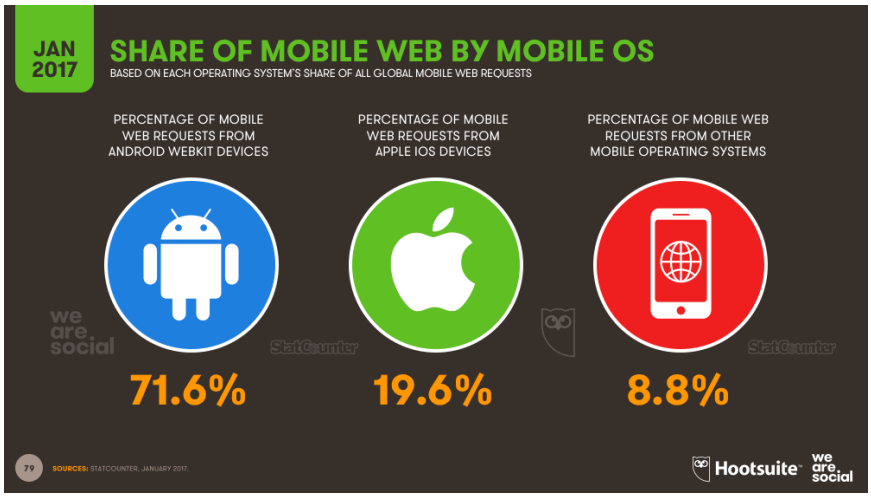

Figura 2.2: Sistemas operacionais móveis mais utilizados em 2017.

2.1.1 A arquitetura do Sistema Android

Como já foi dito anteriormente o Android é um sistema operacional e como qualquer SO é composto por uma pilha de softwares que desempenham papeis específicos dentro do ambiente em que estão sendo executados. O diagrama da Figura 2.3 ilustra a distribuição dos componentes que compõem o Android (DEVELOPERS, 2019i).

A primeira camada da pilha faz referência aos aplicativos que já vem incorporados ao sistema por padrão, são aplicativos que oferecem funcionalidades mais básicas como calendário, relógio e despertador, ligação, agenda, envio de e-mail e SMS, navegador web e a loja de aplicativos que permite que ao usuário acesso a mais de 1.4 milhões de aplicativos.

Uma série de recursos podem ser acessados via APIs (DEVELOPERS, 2019b) oferecidas pelo Android, desenvolvidas na linguagem Java, permitindo que programadores possam desenvolver seus aplicativos de maneira mais simples aproveitando os blocos de código modularizados que já se encontram embarcados no sistema. Alguns exemplos de recursos oferecidos pelas APIs são acesso facilitado a recursos relacionados a interface do usuário como botões, caixas de texto, grades, listas, gerenciador de ciclo de vida dos aplicativos e provedor de acesso a outros aplicativos dos sistema operacional para compartilhar ou consumir dados e gerenciador de notificações.

A maioria dos componentes e serviços oferecidos pelo Android através das APIs Java tem como requisito o uso de bibliotecas nativas em C e C++ que possibilitam a aplicativos de terceiros acesso e utilização desses componentes e serviços. A partir do Android Lollipop (versão 5.0) o sistema passou a contar com o Android Runtime (DEVELOPERS, 2019a) que permite a execução de várias máquinas virtuais com baixo consumo de memória, uma instância para cada processo de aplicativo

em execução. As bibliotecas C e C++ e o ART compõe a terceira camada da arquitetura Android.

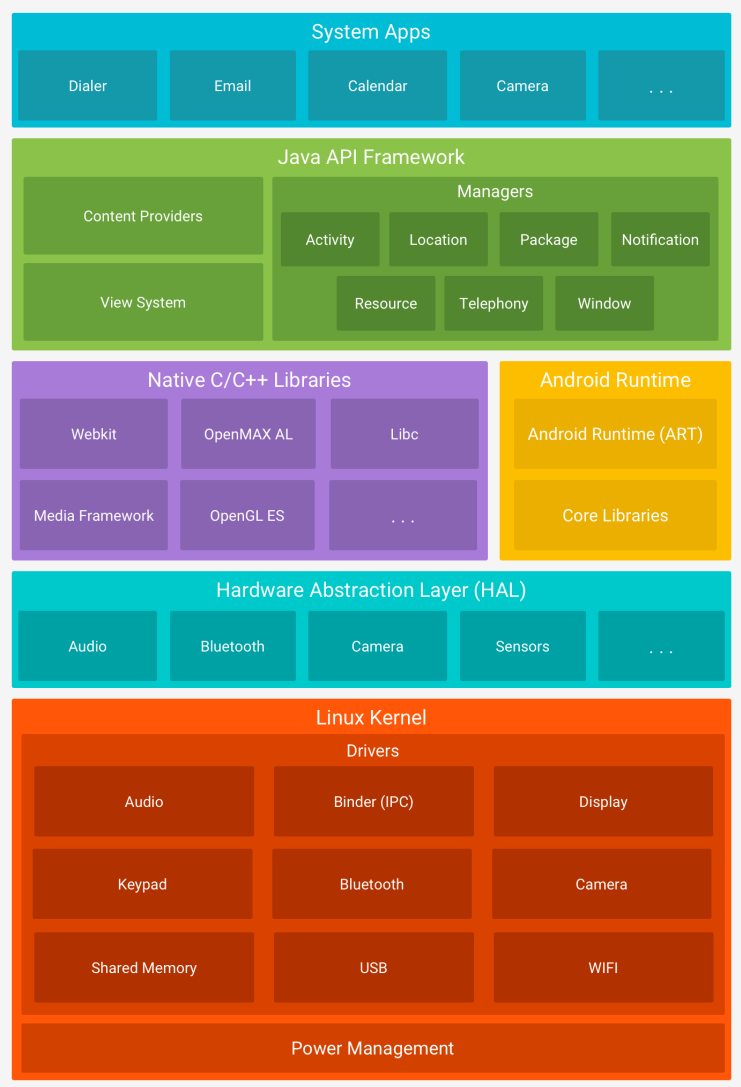

Figura 2.3: Arquitetura do Android.

A quarta camada é composta pela camada de abstração de hardware(HAL) (DEVELOPERS, 2019g) que é responsável por fornecer uma interface de comunicação e acesso entre os hardwares do dispositivo, como câmera e GPS, e as APIs Java citadas anteriormente.

Compondo a quinta e última camada há o kernel Linux (DEVELOPERS, 2019l), também conhecido como o núcleo do sistema, que é a base da plataforma sendo responsável pelo falo de ser a ponte entre o sistema, seus aplicativos e os hardware. È o kernel que executa funções de baixo nível envolvendo, por exemplo, a escrita e leitura na memória e em buffers e a gestão destes recursos.

2.1.2 Os Aplicativos

Os aplicativos para Android são desenvolvidos majoritariamente utilizando a linguagem Java e fazem uso do Software Development Kit (SDK) (DEVELOPERS, 2009) que é um conjunto de ferramentas que conta com bibliotecas de código, debuggers, emulador, exemplos de código e documentação. A distribuição desses aplicativos se dá através dos Android Packages ou APKs que podem ser disponibilizados na Google PlayStore, a loja de aplicativos oficial da Google, ou em lojas de terceiros.

2.1.3 Gerência de memória

Um dos aspectos em que o Android se mostra diferente de outros SOs, em especial os utilizados em computadores desktop e nootebooks, é a maneira de gerenciar os recursos de memória (PHANDROID.COM, 2011). O fato de os dispositivos que carregam esse sistema serem majoritariamente móveis, uma das grandes preocupações é com a gestão de energia da bateria que é consumida principalmente nas operações que envolvem uso memória e iluminação da tela. Assim o Android é capaz de matar processos de aplicações que não estão em uso e impedir o uso de memória das aplicações que estão suspensas.

2.1.4 A Cadeia de Permissões

Um dos padrões de segurança utilizados no Android é a arquitetura de Sandbox(DEVELOPERS, 2019e). Essa arquitetura é utilizada como estratégia para separar os aplicativos uns dos outros e também para limitar o acesso dos mesmos aos recursos do sistema operacional. Neste cenário cada aplicativo roda em uma espécie de container de forma que não é possível para um outro aplicativo malicioso prejudicar os demais ou mesmo o próprio Android.

Contudo não é possível entregar as milhares de soluções existentes no mundo dos dispositivos móveis sem fazer uso de recursos do sistema operacional e do hardware a disposição como GPS, câmera, microfone, interface de rede e bluetooth e muitos outros. Visto que a Sandbox é um ambiente seguro, mas muito limitado, foram desenvolvidas as permissões do sistema(DEVELOPERS, 2019j).

Como já foi dito, por padrão, um aplicativo não tem permissão para acessar informações do sistema operacional e de hardware. Para ganhar acesso a tais recursos é necessário adicionar estaticamente, antes de ser compilado, todas as permissões necessárias para seu correto funcionamento.

As permissões são divididas em dois tipos(DEVELOPERS, 2019h) principais, as ditas normais e as perigosas. Permissões normais são aquelas que, mesmo havendo necessidade do aplicativo manipular dados fora de sua Sandbox, são dados que não comprometem a privacidade do usuário ou prejudicam outros aplicativos e o sistema operacional. Nestes casos o Android concede automaticamente autorização, sem requisitar o usuário. O outro grupo, das permissões perigosas, já abarca todas as permissões que podem comprometer a privacidade e/ou o ambiente onde o aplicativo está inserido. Neste caso o usuário é sempre solicitado para tomar a decisão se deseja permitir ou não

o acesso a tais recursos do sistema.

2.1.5 O Arquivo AndroidManifest

O `AndroidManifest.xml` está contido no código fonte de todo aplicativo disponibilizado na plataforma Google PlayStore. Este arquivo é responsável por centralizar diversas informações que vão desde o nome que aplicativo apresenta ao usuário após ser instalado até às permissões utilizadas para o correto funcionamento dos serviços oferecidos pelo mesmo.

É importante ressaltar, dada a quantidade de frameworks e bibliotecas utilizadas no desenvolvimento de aplicativos mobile para plataformas Android, que a listagem de todas as permissões é independente da tecnologia utilizada para o desenvolvimento do aplicativo, seja ela Java, nativa ou híbrida(TAQTILE, 2016). Em todos os casos a pasta *root* do projeto terá o AndroidManifest.xml.

Algumas das informações contidas no AndroidManifest(DEVELOPERS, 2019c) são:

- Nome do pacote, que é um ID único, e a versão do aplicativo, utilizados durante a compilação e posteriormente pela loja de aplicativos;
- Uma lista de Activity que são módulos responsáveis por abstrair as telas e cada uma de suas funcionalidades;
- Os recievers que compõem a classe responsável por receber informações transmitidas pelo Android ou por outros aplicativos;
- Quais as versões do Android com as quais o aplicativo é compatível;
- Bibliotecas que o aplicativo precisa estar vinculado;
- Tamanhos de tela compatíveis com o aplicativo;
- As permissões de serviço;

Trecho de código 2.1: Declaração de uso de permissões diversas em um arquivo AndroidManifest.xml.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="ACCESS_FINE_LOCATION"/>
<uses-permission android:name="RECORD_AUDIO" />
```

É de vital importância evidenciar que o molde do `AndroidManifest.xml` consta em (**androidmanifest**). Aqui são listadas todas as permissões possíveis que o sistema operacional atende, sendo descrita a permissão, o grupo a qual esta pertence, ícone, descrição e nível de proteção desta.Abaixo é exemplificado a estrutura de uma única permissão (2.2) no 2.3:

Trecho de código 2.2: Exemplo da permissão ACCESS_BACKGROUND_LOCATION

```
<permission android:name="android.permission.ACCESS_FINE_LOCATION"
android:permissionGroup="android.permission-group.UNDEFINED"
android:label="@string/permlab_accessFineLocation"
android:description="@string/permdesc_accessFineLocation"
android:backgroundPermission="android.permission.ACCESS_BACKGROUND_LOCATION"
android:protectionLevel="dangerous|instant" />
```

Trecho de código 2.3: Exemplo de definição de grupo no AndroidManifest.xml.

```
<permission-group android:name="android.permission-group.LOCATION"
android:icon="@drawable/perm_group_location"
android:label="@string/permgrouplab_location"
android:description="@string/permgroupdesc_location"
android:request="@string/permgrouprequest_location"
android:requestDetail="@string/permgrouprequestdetail_location"
android:backgroundRequest="@string/permgroupbackgroundrequest_location"
android:backgroundRequestDetail="@string/permgroupbackgroundrequestdetail_location"
android:priority="400" />
```

A escolha desse grupo de permissões e dessa permissão foi proposital para destacar um ponto importante. Apesar de definido o grupo, nota-se que a permissão que deveria pertencer ao grupo, tem a tag `permissionGroup="android.permission-group.UNDEFINED"`, ou seja, não há grupo atribuído para essa permissão em questão, por mais que seu escopo utilize de recursos que são, em sua essência, de captura de localização.

Cabe aqui então crítica ao `AndroidManifest.xml`, visto que essa condição se repete para várias outras categorias e permissões ao longo do arquivo. Essa é uma observação importante porque o arquivo modelo é utilizado por todos os aplicativos desenvolvidos para a plataforma, como consequência esse padrão é propagado, dificultando a análise para relacionar permissões a seus grupos.

2.1.6 Exemplo de utilização de permissão: A API de localização

O Android utiliza APIs de localização (DEVELOPERS, 2019f) que permitem obter informações sobre o dispositivo e assim prover serviços de rastreamento, reconhecimento geográfico e de atividades automatizadas a partir dos locais que o usuário visita com frequência. É possível obter dados de localização, que em geral são latitude, longitude, data e hora, altitude e velocidade, através de vários sensores presentes nos dispositivos móveis e assim determinar com maior ou menor precisão as coordenadas do usuário.

No geral as informações que são requisitadas via API do sistema são entregues ao processo solicitante através das chamadas *providers* (DEVELOPERS, 2018) que são provedores de conteúdo responsáveis por gerir, encapsular e prover segurança à interface que comunica um o processo requisitante com o bloco de código que executa a operação para entregar os dados de interesse, como ilustrado na Figura 2.5.

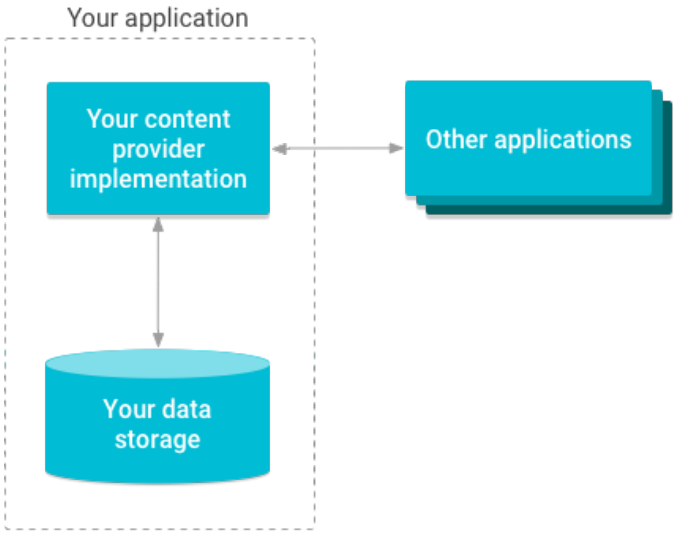

Figura 2.4: Exemplo de provider mediando troca de informações entre aplicação e o SO.

Nas versões mais recentes do Android os aplicativos podem utilizar várias maneiras diferentes de obter a localização do dispositivo, mas em todas elas existem custos envolvidos que devem ser levados em consideração pelo desenvolvedor de forma que atenda a suas necessididades dentro das limitações que em geral são de bateria e tempo de execução. No geral os indicadores utilizados para escolher quais meios utilizar para obter as coordenadas geográficas são o nível de precisão que a aplicação exige e a eficiência da bateria.

Os sinais mais comuns utilizados para obtenção dos dados de localização são GPS, utilizando o sistema de navegação por satélite, o Wi-Fi através de triangulação de pontos de acesso e GSM de maneira semelhante ao o Wi-Fi. Um fator sensível que pode ser configurado de acordo com as demandas da aplicação é o nível de precisão coordenadas:

- _HIGH_ACCURACY: Grande precisão, utiliza todos os sinais disponíveis (GPS, Wi-Fi, GSM e demais sensores que o dispositivo possua). Tem alto dispêndio de energia.
- _BALANCED_POWER_ACCURACY: Tem boa precisão e um consumo energético inferior ao da configuração anterior pelo fato de não usar GPS e sim combinações de Wi-Fi e rede GSM.
- _LOW_POWER: Como o nome sugere este modo consome baixas quantidade de energia e utiliza somente rede GSM, contudo é limitada em relação a precisão da localização que devido a triangulação pode retornar blocos e não um ponto de posicionamento dentro da cidades.
- _NO_POWER: Não utilizada nenhum sinal para calcular localização, apenas aproveitas as informações que já estão na posso de outros aplicativos. Tem consumo de energia zero.

Em relação às permissões, todo e qualquer aplicativo que deseje fazer uso dos recursos de

localização deve ter em ser AndroidManifest.xml as permissões para acessar tais dados. No caso da API de localização as permissões são de dois tipos:

- ACCESS_COARSE_LOCATION: Utiliza precisão grosseira que entrega dados referentes não a um ponto específico, mas a um bloco da cidade;
- ACCESS_FINE_LOCATION: É a permissão utilizada em aplicações que requeiram os resultados mais precisos possíveis.

Sua aplicação segue o modelo do código exemplo:

```
<manifest xmlns: android = "http://schemas.android.com/apk/res/android"
    package = "com.google.android.gms.location.sample.basiclocationsample" >

  <uses-permission android: name = "android.permission.ACCESS_COARSE_LOCATION" />
</ manifest>
```

Capítulo 3

Propostas de solução

Este capítulo apresenta duas propostas para o desenvolvimento de uma solução para monitorar o uso de permissões de serviço em dispositivos Android.

3.1 Abordagem interna: adequação do Android para a entrega dos dados de interesse

Essa abordagem tem como premissa o fato do Android ser um sistema operacional de código aberto. Nesse sentido é possível que usuários terceiros façam alterações no código fonte visando atingir objetivos específicos. Essa abordagem se mostra promissora pelo poder que o desenvolvedor tem de modelar o SO de forma que o mesmo se comporte como desejado. Neste cenário também é objetivo gerar um relatório com os dados investigados. A principal dificuldade neste aqui é a necessidade de lidar diretamente com o sistema operacional promovendo modificações e, por fim, recompilando-o, ambas atividades que demandam mais tempo e conhecimento que a abordagem externa exposta a seguir.

3.2 Abordagem externa: Análise, validação e contestação de permissões

Uma outra opção é o desenvolvimento de uma ferramenta capaz de analisar as permissões de um aplicativo através da engenharia reversa em seu .apk e posterior conversão dos arquivos binários obtidos para leitura e extração de informações de permissão em seu AndroidManifest. De posse das permissões utilizadas pelo aplicativo, é de interesse a contestação destes dados com o que é informado na loja de aplicativos Google PlayStore para que, por fim, seja possível gerar relatórios comparativos entre as permissões informadas e as permissões que de fato são utilizadas pelos aplicativos.

Capítulo 4

Permissões de Serviço e a Ferramenta Desenvolvida

Este capítulo responde algumas perguntas importantes em relação à proposta escolhida para o desenvolvimento da ferramenta, a segurança do AndroidManifest contra possíveis fraudes e discute sobre as possibilidades de injeção de novas permissões após a compilação de aplicativos.

4.1 A Abordagem Escolhida

No Capítulo 3 foram expostas duas propostas de solução para o desenvolvimento de uma ferramenta capaz de verificar o uso de permissões de serviço em sistemas Android. A primeira abordagem mostrou-se muito complexa devido a necessidade de manipular o sistema operacional envolvendo a modificação do código fonte e a recompilação do mesmo. Dentro destas circunstâncias a abordagem externa, que permite a construção de uma ferramenta para análise de aplicativos totalmente fora do sistema operacional, mostrou-se mais promissora e foi escolhida para a implementação da ferramenta.

4.2 AndroidManifest e a Engenharia Reversa

Como foi visto na seção 2.1.5, é no AndroidManifest que as permissões são adicionadas para que os aplicativos tenham acesso a todos os recursos do Android, do hardware e até mesmo de outros aplicativos. Um questionamento natural neste ponto é se o AndroidManifest pode ser alterado por usuários maliciosos após a compilação do .apk. Como já era esperado é sim possível manipular o AndroidManifest após a compilação e conversão do código fonte em .apk.

Existe uma grande variedade de ferramentas de pentest para plataformas Linux, como o Kali Linux, ou mesmo aplicativos Android que são capazes de fazer uma engenharia reversa e através do .apk ter acesso ao código fonte, incluindo o AndroidManifest e a partir deste ponto é possível fazer qualquer modificação que seja de interesse do atacante, inclusive adicionar novas permissões.

Um dos recursos utilizados para evitar este tipo de fraude é a assinatura digital(DEVELOPERS, 2019k) com um certificado que assina o aplicativo no momento de sua compilação de forma que uma nova compilação só possa ser feita caso a chave criada durante a assinatura esteja em um local especificado dentro do código fonte do aplicativo. Contudo, ainda que a assinatura aumente o grau de segurança dos aplicativos, é uma barreira transponível até mesmo por ataques de força bruta, inclusive a própria comunidade cita este recurso de segurança como sendo *easily be bypassed*(GUPTA, 2019).

Com o AndroidManifest e o código fonte disponíveis e vencida a barreira imposta pela assinatura digital, o aplicativo pode facilmente ser manipulado, recompilado e distribuído se passando pelo aplicativo original. Neste cenário é possível, a depender das novas permissões injetadas, que o atacante agora possa obter áudios do ambiente, informações de localização, acesso à câmera, envio de SMS e muito mais, tudo isso até mesmo em background(KOVACS, 2019). Como será visto mais adiante a ferramente desenvolvida também pode ser aplicada em situações onde o AndroidManifest foi fraudado.

4.3 Injeção de Permissões: *Third Party Libraries*

No cenário do desenvolvimento de aplicativos Android existe uma forte tendência para o uso de frameworks e bibliotecas, também conhecidas como *Third Party Libraries*, que permitem entregar soluções cada vez mais rápido devido a alta disponibilidade de funcionalidades prontas e gratuitas oferecidas pela comunidade. Para melhor compreensão podemos utilizar os dados do (NPM, 2018): só dentro do Node Package Menager (NPM), que é um dos gerenciadores de pacotes que podem ser utilizados para desenvolver aplicativos com React Native e Angular, exitem mais de 836 mil bibliotecas disponíveis para uso. Assim é possível baixar e instalar bibliotecas com todas as funcionalidades prontas para usar, por exemplo, a câmera, o gps, leitor de digital , ícones personalizados, modais, conversores de arquivos e milhares de outras funcionalidades.

É indiscutível a importância da comunidade para o desenvolvimento e a manutenção das linguagens e frameworks open source, contudo a utilização de bibliotecas de terceiros tem um problema inerente que é a injeção de permissões no AndroidManifest. Para ilustrar o que pode causar essa injeção de permissões pode-se citar um artigo do blog Medium(MEDIUM, 2019) onde um desenvolvedor de aplicativos Android ilustra, com um caso real, como bibliotecas podem injetar dependências no AndroidManifest.

De acordo com artigo(ROTOLO, 2019), o desenvolvedor recebeu através do sistema de suporte da Google PlayStore a notificação de um usuário do seu aplicativo que estava insatisfeito com o fato do aplicativo, cujo propósito era auxiliar no controle de uma doença, pedir permissão para gravar áudios. Contudo essa permissão não foi inserida no AndroidManifest do aplicativo pelo desenvolvedor e sim por uma biblioteca de terceiros que oferecia a possibilidade de dar feedbacks utilizando notas de áudio. Neste caso tratou-se de uma biblioteca que não era maliciosa, contudo existe sempre a possibilidade de existirem, por exemplo, pacotes com *backdoors* capazes de expor o usuário que utiliza um aplicativo genuíno.

Um outro exemplo de injeção de permissões pode ser visto em um dos app fakes criados neste projeto. Para ter acesso às informações é necessário navegar dentro do diretório do projeto:

```
$ /diretorio_projeto/app/build/outputs/logs/manifest-merger-debug-report.
    txt
```

Este é o arquivo responsável por guardar o log de todas as permissões que são adicionadas ou removidas devido a bibliotecas de terceiros. Como trata-se de um arquivo com centenas de linhas, foi isolada a seguinte parte que indica uma permissão injetada por uma biblioteca externa:

```
ADDED from [:react-native-firebase] /home/paulo/developer/uber-fake/node-
    modules/react-native-firebase/android/build/intermediates/
    library_manifest/debug/AndroidManifest.xml:18:17-116 uses-permission#
    android.permission.ACCESS_WIFI_STATE
```

Trecho de código 4.1: Permissão de acesso à informações da rede WI-FI injetada pela biblioteca react-native-firebase.

4.4 Um Analisador de Permissões - A Engenharia Reversa e o Web Scraping

Percebida a necessidade de avaliar o conjunto de permissões utilizadas por aplicativos Android, foi desenvolvida uma ferramenta capaz de fazer a verificação do AndroidManifest de um .apk e simultaneamente confrontar os resultados obtidos com as informações públicas do respectivo aplicativo na Google PlayStore.

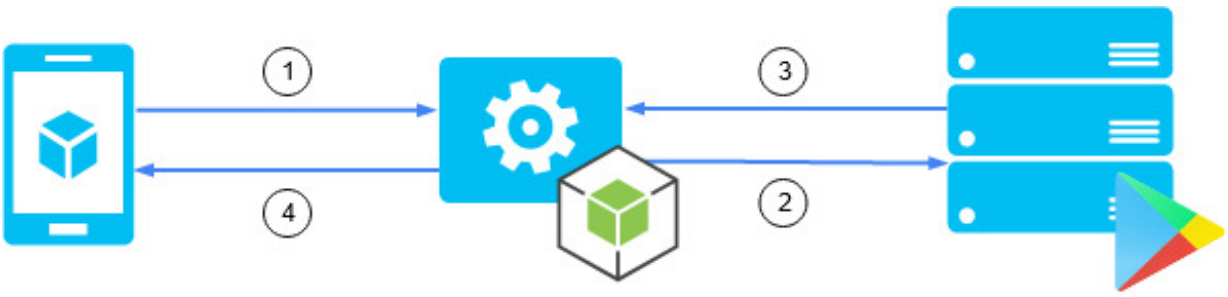

Figura 4.1: Esquemático ilustrando o funcionamento geral da ferramenta.

A API RESTful desenvolvida com NodeJS (NODEJS, 2018), possui duas responsabilidades principais:

1. **Engenharia Reversa do APK**: Neste momento a API recebe via requisição HTTP (POST),

na rota */apk*, o endereço absoluto do aplicativo a ser analisado. Assim é possível ter acesso ao mesmo e iniciar o processo de engenharia reversa.

Este processo consiste passar o .apk pela biblioteca *javascript adm-zip* (CTHACKERS, 2018) que faz uma operação de descompressão expondo o conteúdo do .apk. De posse do diretório resultante, uma função *javascript* varre o diretório até encontrar o arquivo chamado 'AndroidManifest.xml', que é o arquivo de interesse para a análise das permissões utilizadas. Contudo esse arquivo, após a descompressão, está no formato binário.

Para tornar o arquivo legível é feita a última operação do processo de engenharia reversa que consiste em converter o arquivo binário para XML. Ao fim dessa conversão a API devolve como resposta um JSON com as permissões utilizadas pelo aplicativo e explicitadas no AndroidManifest.xml.

```
{ "permissions": [
{ "name": "android.permission.INTERNET" },
{ "name": "android.permission.SYSTEM_ALERT_WINDOW" },
{ "name": "android.permission.CAMERA" },
{ "name": "android.permission.WRITE_EXTERNAL_STORAGE" },
{ "name": "android.permission.ACCESS_FINE_LOCATION" },
{ "name": "android.permission.RECORD_AUDIO" },
{ "name": "android.permission.READ_PHONE_STATE" },
{ "name": "android.permission.READ_EXTERNAL_STORAGE" }
]}
```

Trecho de código 4.2: Exemplo de resposta da API NodeJS à requisição na rota /apk, passando como parâmetro no *body* da requisição POST o endereço local do aplicativo na árvore de arquivos do computador utilizado para realizar a bateria de testes.

2. **Operação de *scrap* na Google PlayStore**: Para obter informações referentes ao aplicativo desejado na Google Play Store é utilizado, também pela API NodeJS, operações de Web scraping que utilizam palavras-chave nas suas requisições para pesquisar pelos aplicativos, pacotes e suas respectivas permissões.

   Este segundo estágio é mais simplificado que a fase de engenharia reversa do ponto de vista da quantidade de código escrito, contudo requer um processo de duas etapas. Isso se dá pelo fato das operações de scrap não serem sempre totalmente eficientes em relação aos resultados obtidos com as pesquisas. Por isso as requisições são divididas em duas etapas.

   Inicialmente o cliente pesquisa pelo nome comercial do aplicativo que deseja conhecer as permissões utilizadas. Neste momento a pesquisa é via requisição HTTP GET passando como parâmetro o nome do aplicativo na URL da requisição na rota */app* como pode ser visto abaixo. É importante ressaltar também que o retorno da consulta, de acordo com o parametrizado na API, são de 5 resultados similares ao parâmetro de busca, para simplificação do exemplo, o trecho abaixo conta apenas com o resultado mais relevante para exposição de seu funcionamento.

```
{ "resultado": [
    {
      "title": "Banco do Brasil",
      "appId": "br.com.bb.android",
      "url": "https://play.google.com/store/apps/details?id=br.com.bb.android
          ",
      "icon": "https://lh3.googleusercontent.com/
          lwyFOE-EIz6r9flTQ_bJVVnm5uMaMRgD1G1x4TlHqBYbK2uR9oB5jwoAdxqEVS41fTM
          ",
      "developer": "Banco do Brasil SA",
      "developerId": "Banco+do+Brasil+SA",
      "priceText": "FREE",
      "free": true,
      "summary": "#MaisQueDigital is to have the entire bank with just one
          touch.",
      "scoreText": "4.6",
      "score": 4.581957
    },
    ...
```


Trecho de código 4.3: Retorno do método HTTP GET pelo aplicativo "Banco do Brasil"disponível na PlayStore, feito à API NodeJS de domínio */app* uma resposta HTTP 200 com os parâmetros apresentados. Os demais resultados da consulta foram omitidos.

Analisando o primeiro objeto da resposta, que contém os dados do aplicativo "Banco do Brasil", é possível notar que uma das chaves, a "appID", traz o nome do pacote, que é um identificador único utilizado pela loja de aplicativos para diferenciar cada APK. Este identificador é o parâmetro utilizado no próximo passo para acesso às permissões utilizadas pelo aplicativo em questão.

De posse do appID é necessário fazer uma nova requisição HTTP (POST) à API NodeJS passando no *body* da requisição um parâmetro chamado "appID"com o nome do pacote. Como resposta a API responde com um JSON contendo todas as permissões utilizadas pelo aplicativo assim como uma pequena descrição para cada uma delas, segundo o desenvolvedor e a própria Google PlayStore.


```
{
  "resultado": [
    {
      "permission": "find accounts on the device",
      "type": "Identity"
    },
    {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
    },
    {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    },
    {
```

```
16        "permission": "take pictures and videos",
17        "type": "Camera"
18      },
19      {
20        "permission": "read phone status and identity",
21        "type": "Device ID & call information"
22      },
23      {
24        "permission": "modify or delete the contents of your USB storage",
25        "type": "Storage"
26      },
27
28      ...
29
30      }
```

Trecho de código 4.4: Ao passar o nome do pacote no *body* da requisição é obtido como resposta um objeto com todas as permissões utilizadas pelo aplicativo de acordo com a Google PlayStore. Neste caso o aplicativo estava utilizando mais de trinta e quatro permissões de acordo com a loja. Para melhor visualização os primeiro seis objetos foram abertos para expor seus detalhes.

4.5 O Comparador

Ao ter acesso às informações provenientes da análise do AndroidManifest do aplicativo e do *scraper* na Google PlayStore é possível fazer algumas comparações entre as informações contidas na loja e o real uso das permissões. Contudo, para facilitar esta análise foi desenvolvido um terceiro módulo chamado de Comparador.

O módulo Comparador tem como objetivo central a discriminação das permissões e sua separação em grupos, cada grupo representa um tipo de permissão. Cada um dos grupos veio de uma análise minuciosa dos retornos a consultas feitas pelo *scraper* na loja de aplicativos. Assim é possível ter um entendimento claro sobre quais permissões estão sendo utilizadas e a quais grupos as mesmas pertencem.

Essas informações são suficientes para mostrar a quais tipos de vulnerabilidade o usuário e suas informações estão submetidos. Por fim ainda é obtido um grupo de permissões chamado de "Outros", este grupo reúne todas as permissões que não são claramente classificadas pela loja, mas que também podem oferecer riscos ao usuário. A seguir é apresentada a lista de grupos e uma pequena descrição das suas permissões.

- ACTIVITY_RECOGNITION: Utilizada para detecção de atividades do usuário como uma corrida, caminhada ou alguma outra atividade específica que envolve o deslocamento do dispositivo.
  - ACTIVITY_RECOGNITION
- CALENDAR: Permite a ler e escrever através da API de Calendário do sistema Android.
  - READ_CALENDAR
  - WRITE_CALENDAR
- CALL_LOG: Acesso completo, leitura e escrita, a todas as informações referentes às ligações feitas e recebidas através do dispositivo Android.
  - ACCESS_IMS_CALL_SERVICE
  - PROCESS_OUTGOING_CALLS
  - READ_CALL_LOG
  - WRITE_CALL_LOG
- CAMERA: Fornece acesso ao hardware de câmera e microfone. Possibilita a utilização das funções foto e vídeo e ainda a gravação de áudio ambiente.
  - CAMERA
- CONTACTS: Acesso completo aos contatos gravados no banco de dados local do Android.
  - READ_CONTACTS
  - WRITE_CONTACTS
- LOCATION: Fornecesse acesso à API de localização do Android através de GPS ou sincronização com redes de dados móveis ou Wi-Fi. Ainda é possível escolher, de acordo com a permissão utilizada, o nível de precisão dos dados de localização aferidos.
  - ACCESS_BACKGROUND_LOCATION
  - ACCESS_COARSE_LOCATION
  - ACCESS_FINE_LOCATION
  - ACCESS_LOCATION_EXTRA_COMMANDS
- MICROPHONE: Permite acesso ao microfone do dispositivo para gravação de áudios locais.
  - RECORD_AUDIO
- PHONE: É através das permissões deste grupo que um aplicativo recebe autorização para gerenciar informações relacionadas a utilização da rede celular.
  - ACCEPT_HANDOVER
  - ACCESS_UCE_PRESENCE_SERVICE

  - ACCESS_UCE_OPTIONS_SERVICE
  - ADD_VOICEMAIL
  - ANSWER_PHONE_CALLS
  - CALL_COMPANION_APP
  - CALL_PHONE
  - MANAGE_OWN_CALLS
  - READ_PHONE_STATE
  - RED_PHONE+NUMBERS
  - USE_SIP

- SENSORS: Reúne permissões relacionadas aos diversos sensores e às APIs de reconhecimento biométrico.
  - BODY_SENSORS
  - USE_BIOMETRIC
  - USE_FINGERPRINT

- STORAGE: Acesso ao hardware de armazenamento/persistência do dispositivo. Oferece opções de leitura e escrita em disco.
  - ACCESS_MEDIA_LOCATION
  - READ_EXTERNAL_STORAGE
  - WRITE_EXTERNAL_STORAGE
  - WRITE_OBB

- NETWORK: São permissões que permitem que aplicativos acessem uma gama de informações sobre as redes às quais o dispositivo se conecta.
  - INTERNET
  - ACCESS_NETWORK_STATE
  - ACCESS_WIFI_STATE
  - CHANGE_WIFI_STATE
  - MANAGE_IPSEC_TUNNELS
  - MANAGE_TEST_NETWORKS
  - BLUETOOTH
  - SUSPEND_APPS
  - BLUTOOTH_ADMIN
  - BLUETOOTH_PRIVILEGED
  - BLUETOOTH_MAP

  - BLUETOOTH_STACK
  - NFC
  - NFC_TRANSACTION_EVENT

- ACCOUNT: Oferece acesso às contas existentes no dispositivo, além do tipo de conta e a qual usuário está vinculada. Também é possível ter acesso a quais recursos estão ativados para cada conta.
  - GET_ACCOUNTS

- OUTROS: Reúne todas as permissões que não são explicitamente classificadas pela Google PlayStore, mas que estão em uso pelos aplicativos e por isso também representam riscos para a segurança dos dados do usuário.

4.6 A Interface Web

Após a construção da ferramenta, e com seus três estágios em funcionamento - web scrap, engenharia reversa e o comparador - e com os métodos e retornos devidamente apresentados, foi desenvolvida uma interface web para oferecer uma melhor experiência de usuário. Utilizando o *framework* Angular 8 (CALADO, 2019).

Para justificar a escolha do Angular 8 vale citar alguns aspectos que constituem sus história de formação e algumas de suas características como ferramenta para desenvolvimento de interfaces. Este *framework* é derivado do legado AngularJS, que surgiu em 2010 como concorrente dos já populares *Vue.js* e *React.js* no desenvolvimento de *Single Page Applications* . Implementando o *superset* JavaScript nomeado de TypeScript, o Angular está entre os *frameworks* mais usados no mundo principalmente para o desenvolvimento de aplicações *front-end* SPA (*Single Page Application*), devido a sua rápida implementação e pouco dependente de *plugins* ou aditivos de terceiros.

A a SPA desenvolvida para este projeto conta com cinco *Components*, compostos de um núcleo de *scripts* em TypeScript, exibição em HTML e CSS para estilização. Utiliza quatro *Services* que consomem os *endpoints* oferecidos pela API NodeJS e quatro *Models* que capturam e tratam os resultados das consultas à API, além de padronizam suas entradas.

A arquitetura da SPA desenvolvida pode ser definida a partir do diagrama de blocos a seguir:

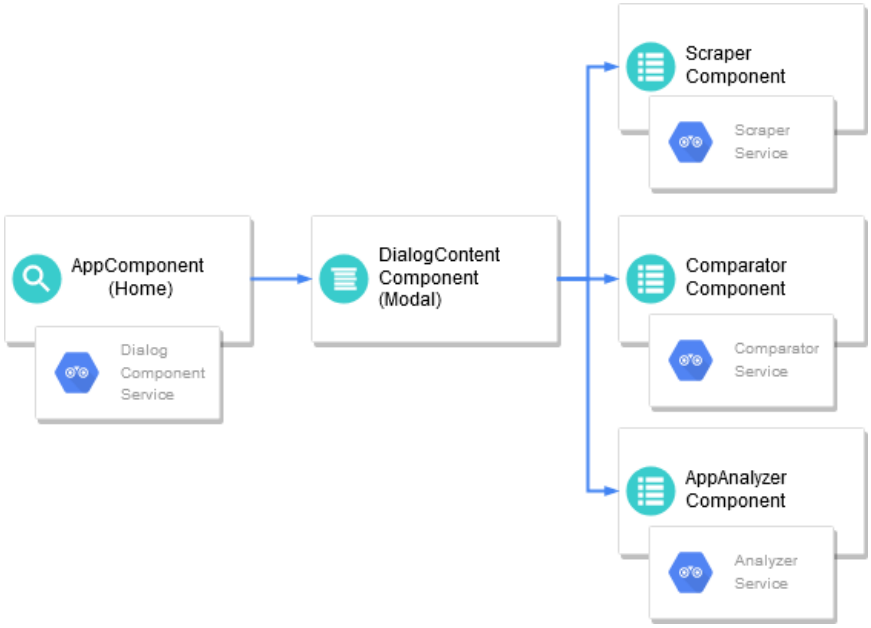

Figura 4.2: Diagramas de blocos que define a estrutura da SPA desenvolvida como interface Web para exibição dos dados coletados.

Ainda sobre a interface Web desenvolvida, as duas imagens abaixo ilustram o estilo e disposição dos itens nas páginas elaboradas (figuras 4.3 e 4.4):

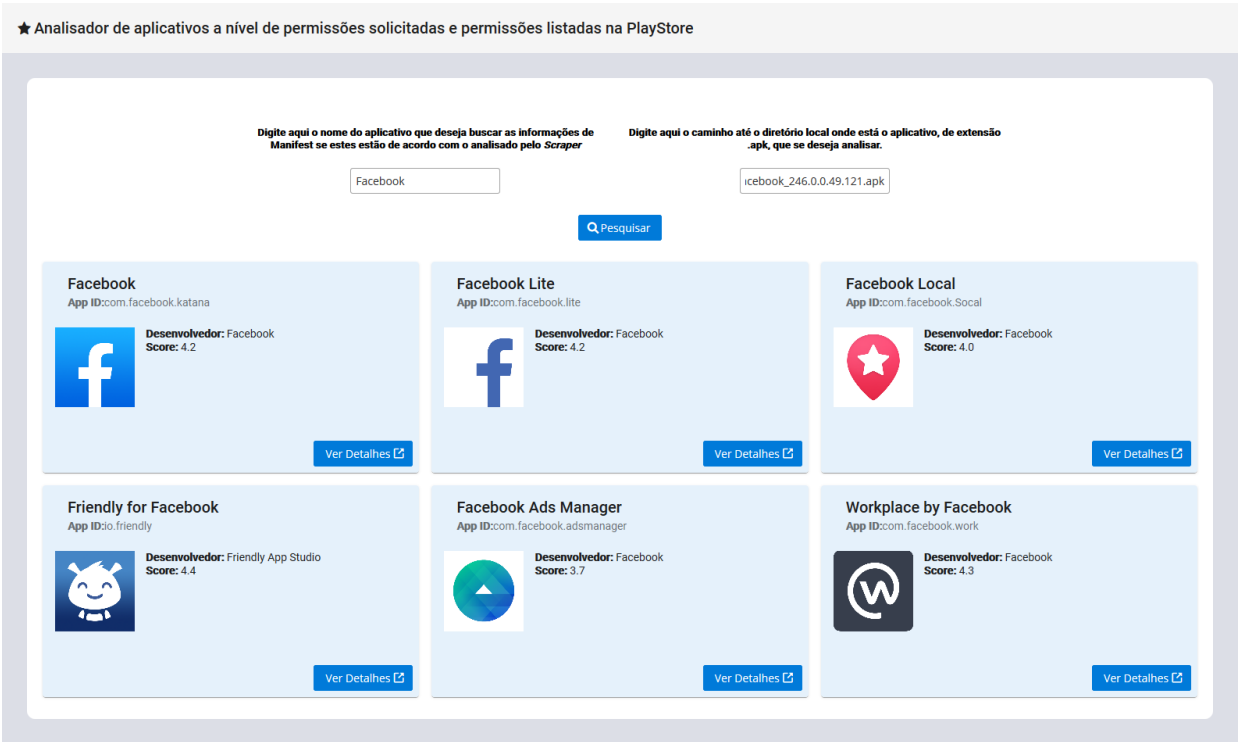

Figura 4.3: Página inicial da interface web após a busca pelo aplicativo "Facebook". Neste passo inicial é passando nos campos de pesquisa a *string* "Facebook", à esquerda, e o diretório onde o APK do Facebook se encontra à direita.

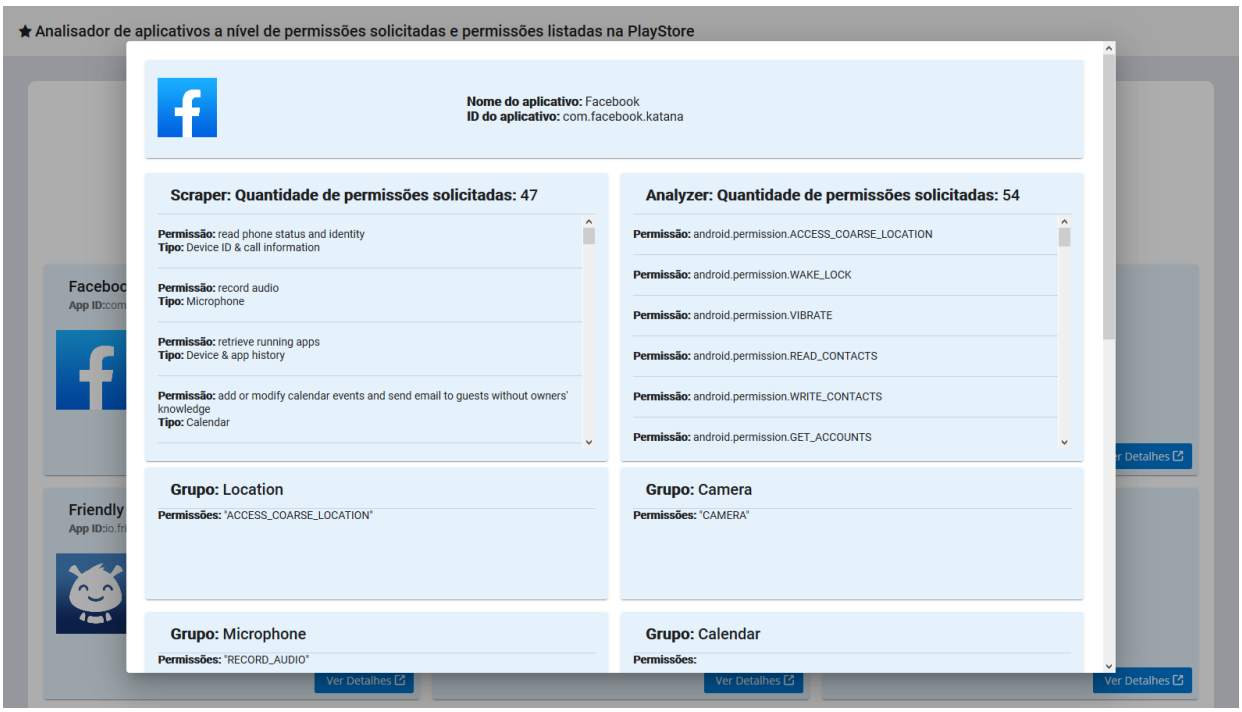

Figura 4.4: Como resultado das pesquisas da Figura 4.4 é obtido uma lista de permissões utilizadas pelo aplicativo de acordo com o AndroidManifest, à direita, e uma lista das permissões informadas pela Google PlayStore à esquerda. Ao final é possível ver alguns grupos e quais permissões estão associadas a eles.

Capítulo 5

Análise de Resultados

Esse capítulo descreve os testes feitos utilizando a ferramenta, detalhando as etapas da implementação, as principais vantagens e limitações.

5.1 Consistência da Solução

Na seção 4.2 foi explicada uma das estratégias utilizadas por atacantes para fraudar o AndroidManifest de aplicativos Android. Neste contexto foram desenvolvidos testes para verificar a consistência da ferramenta diante de situações em que ocorre o tipo de fraude citada via engenharia reversa. Para implementar os testes foi utilizado o aplicativo AppINC (LABFORENSE, 2019), desenvolvido pelo Labforense - ENE/UNB em parceria com o Instituto Nacional de Criminalística da Polícia Federal. Essa escolha foi feita pelo fato dos autores deste projeto possuírem a chave de compilação do aplicativo, o que permitiu pular a etapa de decifração da mesma. Vale ressaltar que essa escolha não trouxe nenhum impacto para os resultados obtidos. Esta etapa foi abstraída do processo pelo fato de a quebra das chaves de compilação dos aplicativos serem uma atividade fora do escopo do projeto.

Uma versão fake do AppINC, o AppINCFake foi criada. Todo o código se manteve inalterado, exceto por uma modificação pontual. Foram adicionadas, antes da compilação da versão fake, novas permissões dando acesso a leitura de SMS e acesso à informações de localização em *background* para simular uma fraude no `AndroidManifest` de um aplicativo da Google PlayStore. Após a compilação da versão fake, ambos tiveram suas permissões analisadas pela ferramenta. Como já havia sido dito no capítulo 4, a API NodeJS foi capaz de mostrar as modificações que haviam sido feitas no `AndroidManifest` do aplicativo em questão.

```
"permissoes":[
    { "name" : "android.permission.INTERNET" },
    { "name" : "android.permission.SYSTEM_ALERT_WINDOW" },
    { "name" : "android.permission.CAMERA" },
    { "name" : "android.permission.WRITE_EXTERNAL_STORAGE" },
    { "name" : "android.permission.ACCESS_FINE_LOCATION" },
    { "name" : "android.permission.RECORD_AUDIO" },
```

```
8       { "name" : "android.permission.READ_PHONE_STATE" },
9       { "name" : "android.permission.READ_EXTERNAL_STORAGE" }
10  ]
```

Trecho de código 5.1: Resultado da análise do AppINC genuino feita pela API NodeJS.

```
1   "permissoes":[
2       { "name" : "android.permission.INTERNET" },
3       { "name" : "android.permission.SYSTEM_ALERT_WINDOW" },
4       { "name" : "android.permission.CAMERA" },
5       { "name":  "android.permission.READ_SMS"} ,
6       { "name" : "android.permission.WRITE_EXTERNAL_STORAGE" },
7       { "name":  "android.permission.ACCESS_BACKGROUND_LOCATION"} ,
8       { "name" : "android.permission.ACCESS_FINE_LOCATION" },
9       { "name" : "android.permission.RECORD_AUDIO" },
10      { "name" : "android.permission.READ_PHONE_STATE" },
11      { "name" : "android.permission.READ_EXTERNAL_STORAGE" }
12  ]
```

Trecho de código 5.2: Resultado da análise do AppINCFake mostrando, em destaque, duas novas permissões.

Numa situação real em que ocorresse uma modificação no `AndroidManifest` de um aplicativo semelhante a esta, o atacante poderia ter acesso a leitura de mensagens SMS do dispositivo, além de poder, a qualquer momento, ler informações acerca da localização do dispositivo, totalmente em *background*. Este exemplo ilustra o quão vulnerável o usuário pode ficar a depender do tipo de permissão que seus aplicativos estão utilizando.

5.2 Testes e Análise dos Resultados

Validado a consistência da solução apresentada, exibindo a capacidade de detectar a injeção de permissões não listadas na PlayStore sobre dado aplicativo, resta então apresentar alguns casos de uso da ferramenta, e realizar uma avaliação crítica levando em consideração a motivação e proposta inicial deste projeto: analisar as permissões listadas na Google PlayStore e verificar se estas estão em acordo com as permissões solicitadas no `AndroidManifest` do aplicativo.

Serão utilizados cinco aplicativos comerciais, disponibilizados para download na Google PlayStore:

- GMail;
- OLX;
- Instagram;
- Caixa;
- WhatsApp;

Cada um desses aplicativos, com mais de 1 milhão de downloads registrados, serão objeto de estudo e análise a partir da ferramenta elaborada.

5.2.1 GMail — appId: com.google.android.gm


```
{ "resultado": [{
      "permission": "record audio",
      "type": "Microphone"
    }, {
      "permission": "read calendar events plus confidential information",
      "type": "Calendar"
    }, {
      "permission": "add or modify calendar events and send email to guests
         without owners' knowledge",
      "type": "Calendar"
    }, {
      "permission": "take pictures and videos",
      "type": "Camera"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "read your contacts",
      "type": "Contacts"
    }, {
      "permission": "find accounts on the device",
      "type": "Contacts"
    }, {
      "permission": "modify your contacts",
      "type": "Contacts"
    }, {
      "permission": "find accounts on the device",
      "type": "Identity"
    }, {
      "permission": "add or remove accounts",
      "type": "Identity"
    }, {
      "permission": "read your own contact card",
      "type": "Identity"
    }, {
      "permission": "write call log",
```

```
      "type": "Phone"
    }, {
      "permission": "read call log",
      "type": "Phone"
    }, {
      "permission": "view Wi-Fi connections",
      "type": "Wi-Fi connection information"
    }, {
      "permission": "toggle sync on and off",
      "type": "Other"
    }, {
      "permission": "create accounts and set passwords",
      "type": "Other"
    }, {
      "permission": "change your audio settings",
      "type": "Other"
    }, {
      "permission": "install shortcuts",
      "type": "Other"
    }, {
      "permission": "pair with Bluetooth devices",
      "type": "Other"
    }, {
      "permission": "read sync settings",
      "type": "Other"
    }, {
      "permission": "use accounts on the device",
      "type": "Other"
    }, {
      "permission": "prevent device from sleeping",
      "type": "Other"
    }, {
      "permission": "read Google service configuration",
      "type": "Other"
    }, {
      "permission": "view network connections",
      "type": "Other"
    }, {
      "permission": "measure app storage space",
      "type": "Other"
    }, {
      "permission": "run at startup",
      "type": "Other"
    }, {
      "permission": "full network access",
      "type": "Other"
    }, {
      "permission": "control vibration",
      "type": "Other"
    }, {
      "permission": "control Near Field Communication",
```

```
97        "type": "Other"}]}
```

Trecho de código 5.3: Solicitação de permissões listadas listadas na PlayStore para o aplicativo GMail, capturadas pelo módulo Scraper.

```
1  Group: Accounts
2  Permissions: { "MANAGE_ACCOUNTS","AUTHENTICATE_ACCOUNTS" }
3  Group: Location
4  Permissions: { "ACCESS_FINE_LOCATION","ACCESS_COARSE_LOCATION" }
5  Group: Camera
6  Permissions: { "CAMERA" }
7  Group: Microphone
8  Permissions: {}
9  Group: Calendar
10 Permissions: { "READ_CALENDAR","WRITE_CALENDAR" }
11 Group: Phone
12 Permissions: { "READ_PHONE_STATE" }
13 Group: Contacts
14 Permissions: { "READ_CONTACTS","android.permission.WRITE_CONTACTS" }
15 Group: Storage
16 Permissions: { "WRITE_EXTERNAL_STORAGE","READ_EXTERNAL_STORAGE" }
17 Group: Sensors
18 Permissions: { "USE_FINGERPRINT" }
19 Group: SMS
20 Permissions: {}
21 Group: Call log
22 Permissions: {}
23 Group: Outros
24 Permissions: { "BARCODE_SCANNED","VIBRATE","ACTION_MANAGE_OVERLAY_PERMISSION","
       SYSTEM_ALERT_WINDOW","GET_ACCOUNTS","BIND_NOTIFICATION_LISTENER_SERVICE","
       RECEIVE_BOOT_COMPLETED","FOREGROUND_SERVICE","INSTALL_SHORTCUT","
       READ_SYNC_SETTINGS","READ_SYNC_STATS","SUBSCRIBED_FEEDS_READ","
       SUBSCRIBED_FEEDS_WRITE","USE_CREDENTIALS","VIBRATE","WAKE_LOCK","
       WRITE_SYNC_SETTINGS","READ_OOBE","READ_GMAIL","WRITE_GMAIL","GOOGLE_AUTH","
       GOOGLE_AUTH.mail","READ_GSERVICES","AUTO_SEND","AUDIO_FILE_ACCESS","
       DOWNLOAD_WITHOUT_NOTIFICATION","READ_ATTACHMENT","ACCESS_PROVIDER","
       READ_PROFILE","GET_WIDGET_UPDATE","BROADCAST_INTERNAL","BIND","
       PARTNER_PROVIDER","GET_PACKAGE_SIZE","REQUEST_SCREEN_LOCK_COMPLEXITY","RECEIVE
       " }
25 Group: Network
26 Permissions: { "INTERNET","ACCESS_NETWORK_STATE","ACCESS_WIFI_STATE" }
```

Trecho de código 5.4: Permissões solicitadas pelo aplicativo GMail em seu AndroidManifest, avaliado pelo módulo Analisador, e devidamente alocados em seus determinados grupos.

O aplicativo GMail listou um total de 32 solicitações de permissões na PlayStore, porém, ao avaliar seu `AndroidManifest.xml`, foram observadas 42 permissões, ao agrupá-las utilizando o módulo Analisador.

5.2.2 OLX

```
1 {"resultado": [{
```

```
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "take pictures and videos",
      "type": "Camera"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "view Wi-Fi connections",
      "type": "Wi-Fi connection information"
    }, {
      "permission": "precise location (GPS and network-based)",
      "type": "Location"
    }, {
      "permission": "approximate location (network-based)",
      "type": "Location"
    }, {
      "permission": "connect and disconnect from Wi-Fi",
      "type": "Other"
    }, {
      "permission": "read Google service configuration",
      "type": "Other"
    }, {
      "permission": "prevent device from sleeping",
      "type": "Other"
    }, {
      "permission": "run at startup",
      "type": "Other"
    }, {
      "permission": "full network access",
      "type": "Other"
    }, {
      "permission": "view network connections",
      "type": "Other"
    }, {
      "permission": "control vibration",
      "type": "Other"
    }]}
```

Trecho de código 5.5: Solicitação de permissões listadas na PlayStore para o aplicativo OLX, capturadas pelo módulo Scraper.

```
Group: Location
Permissions: { "ACCESS_FINE_LOCATION","ACCESS_COARSE_LOCATION" }
Group: Camera
```

```
Permissions: { "CAMERA" }
Group: Microphone
Permissions: {}
Group: Calendar
Permissions: {}
Group: Phone
Permissions: {}
Group: Contacts
Permissions: {}
Group: Storage
Permissions: { "READ_EXTERNAL_STORAGE","WRITE_EXTERNAL_STORAGE" }
Group: Sensors
Permissions: {}
Group: SMS
Permissions: {}
Group: Call log
Permissions: {}
Group: Outros
Permissions: { "FOREGROUND_SERVICE","RECEIVE_BOOT_COMPLETED","READ_GSERVICES","
    UA_DATA","RECEIVE","WAKE_LOCK","VIBRATE","C2D_MESSAGE","
    BIND_GET_INSTALL_REFERRER_SERVICE","READ_GSERVICES","UA_DATA","RECEIVE","
    WAKE_LOCK","VIBRATE","C2D_MESSAGE","ACTIVITY_RECOGNITION","
    BIND_GET_INSTALL_REFERRER_SERVICE" }
Group: Network
Permissions: { "ACCESS_NETWORK_STATE","CHANGE_WIFI_STATE","ACCESS_WIFI_STATE","
    INTERNET" }
```

Trecho de código 5.6: Permissões solicitadas pelo aplicativo OLX em seu AndroidManifest, avaliado pelo módulo Analisador, e devidamente alocados em seus determinados grupos.

A proposta da OLX se assemelha a um fórum baseado em tópicos, onde há comunicação apenas entre o criador do tópico e os interessados no produto a venda.

Apesar de utilizar permissões próprias como a *androidApp .permission.UA_DATA*, que não são muito bem documentadas, o aplicativo conta com 15 solicitações de permissão informadas pela Google PlayStore e 19 efetivamente utilizadas. Esse comportamento gera suspeitas visto que ao utilizar permissões como *READ_CALL_LOG*, utilizada para que o aplicativo leia os registros de chamadas feitos pelo dispositivo, o aplicativo está saindo do escopo da sua proposta central que é compra e venda online.

5.2.3 Instagram

```
{"resultado": [{
      "permission": "read your own contact card",
      "type": "Identity"
    }, {
      "permission": "find accounts on the device",
      "type": "Identity"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
```

```
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "take pictures and videos",
      "type": "Camera"
    }, {
      "permission": "read phone status and identity",
      "type": "Device ID & call information"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "retrieve running apps",
      "type": "Device & app history"
    }, {
      "permission": "directly call phone numbers",
      "type": "Phone"
    }, {
      "permission": "read phone status and identity",
      "type": "Phone"
    }, {
      "permission": "find accounts on the device",
      "type": "Contacts"
    }, {
      "permission": "read your contacts",
      "type": "Contacts"
    }, {
      "permission": "record audio",
      "type": "Microphone"
    }, {
      "permission": "receive text messages (SMS)",
      "type": "SMS"
    }, {
      "permission": "view Wi-Fi connections",
      "type": "Wi-Fi connection information"
    }, {
      "permission": "precise location (GPS and network-based)",
      "type": "Location"
    }, {
      "permission": "change your audio settings",
      "type": "Other"
    }, {
      "permission": "install shortcuts",
      "type": "Other"
    }, {
      "permission": "access Bluetooth settings",
      "type": "Other"
    }, {
```

```
      "permission": "use accounts on the device",
      "type": "Other"
    }, {
      "permission": "connect and disconnect from Wi-Fi",
      "type": "Other"
    }, {
      "permission": "prevent device from sleeping",
      "type": "Other"
    }, {
      "permission": "uninstall shortcuts",
      "type": "Other"
    }, {
      "permission": "run at startup",
      "type": "Other"
    }, {
      "permission": "full network access",
      "type": "Other"
    }, {
      "permission": "read battery statistics",
      "type": "Other"
    }, {
      "permission": "view network connections",
      "type": "Other"
    }, {
      "permission": "pair with Bluetooth devices",
      "type": "Other"
    }, {
      "permission": "control vibration",
      "type": "Other"}]}
```

Trecho de código 5.7: Solicitação de permissões listadas na PlayStore para o aplicativo Instagram, capturadas pelo módulo Scraper.

```
Group: Location
Permissions: { "ACCESS_FINE_LOCATION" }
Group: Camera
Permissions: { "CAMERA" }
Group: Microphone
Permissions: { "RECORD_AUDIO" }
Group: Calendar
Permissions: {}
Group: Phone
Permissions: { "READ_PHONE_STATE" }
Group: Contacts
Permissions: { "READ_CONTACTS" }
Group: Storage
Permissions: { "WRITE_EXTERNAL_STORAGE" }
Group: Sensors
Permissions: {}
Group: SMS
Permissions: {}
Group: Call log
```

```
Permissions: {}
Group: Outros
Permissions: { "WAKE_LOCK","GET_ACCOUNTS","USE_CREDENTIALS","
    RECEIVE_BOOT_COMPLETED","VIBRATE","FOREGROUND_SERVICE","RECEIVE","READ_PROFILE
    ","MODIFY_AUDIO_SETTINGS","INSTALL_SHORTCUT","UNINSTALL_SHORTCUT","
    PROTECTED_DEEPLINKING","DIRECT_APP_THREAD_STORE_SERVICE","READ_SETTINGS","
    UPDATE_SHORTCUT","CHANGE_BADGE","BROADCAST_BADGE","C2D_MESSAGE","
    RECEIVE_ADM_MESSAGE","RECEIVE" }
Group: Network
Permissions: { "INTERNET","ACCESS_NETWORK_STATE","INTERNET","ACCESS_NETWORK_STATE
    " }
```

Trecho de código 5.8: Permissões solicitadas pelo aplicativo Instagram em seu AndroidManifest, avaliado pelo módulo Analisador, e devidamente alocados em seus determinados grupos.

As redes sociais são os principais aplicativos consumidos no ambiente *mobile* e, dado a relevância desse tipo de aplicativo no cenário atual, é importante avaliar quais acessos são solicitados por esse tipo de ferramenta. Com 30 solicitações de permissão informadas e 28 solicitadas, a plataforma de compartilhamento de fotos e vídeos está bem atrás de maior concorrente, o aplicativo Facebook, que informa ao usuário 47 permissões solicitadas ao instalar e utiliza efetivamente 54 permissões do sistema, sendo destas 54, 48 somente do sistema operacional.

5.2.4 Caixa

```
{"resultado": [{
      "permission": "find accounts on the device",
      "type": "Identity"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "take pictures and videos",
      "type": "Camera"
    }, {
      "permission": "read phone status and identity",
      "type": "Device ID & call information"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read phone status and identity",
      "type": "Phone"
    }, {
      "permission": "find accounts on the device",
      "type": "Contacts"
```

```
    }, {
      "permission": "read your contacts",
      "type": "Contacts"
    }, {
      "permission": "view Wi-Fi connections",
      "type": "Wi-Fi connection information"
    }, {
      "permission": "precise location (GPS and network-based)",
      "type": "Location"
    }, {
      "permission": "approximate location (network-based)",
      "type": "Location"
    }, {
      "permission": "run at startup",
      "type": "Other"
    }, {
      "permission": "draw over other apps",
      "type": "Other"
    }, {
      "permission": "full network access",
      "type": "Other"
    }, {
      "permission": "view network connections",
      "type": "Other"
    }, {
      "permission": "control vibration",
      "type": "Other"}]}
```

Trecho de código 5.9: Solicitação de permissões listadas listadas na PlayStore para o aplicativo CAIXA, capturadas pelo módulo Scraper.

```

Group: Location
Permissions: { "ACCESS_FINE_LOCATION","ACCESS_COARSE_LOCATION" }
Group: Camera
Permissions: { "CAMERA" }
Group: Microphone
Permissions: {}
Group: Calendar
Permissions: {}
Group: Phone
Permissions: { "READ_PHONE_STATE" }
Group: Contacts
Permissions: { "READ_CONTACTS" }
Group: Storage
Permissions: { "WRITE_EXTERNAL_STORAGE","READ_EXTERNAL_STORAGE" }
Group: Sensors
Permissions: { "USE_FINGERPRINT" }
Group: SMS
Permissions: {}
Group: Call log
Permissions: {}
```

```
Group: Outros
Permissions: { "BARCODE_SCANNED","VIBRATE","ACTION_MANAGE_OVERLAY_PERMISSION","
    SYSTEM_ALERT_WINDOW","GET_ACCOUNTS","BIND_NOTIFICATION_LISTENER_SERVICE","
    RECEIVE_BOOT_COMPLETED" }
Group: Network
Permissions: { "INTERNET","ACCESS_NETWORK_STATE","ACCESS_WIFI_STATE" }
```

Trecho de código 5.10: Permissões solicitadas pelo aplicativo CAIXA em seu AndroidManifest, avaliado pelo módulo Analisador, e devidamente alocados em seus determinados grupos.

A sensibilidade da informação bancária dos usuários é levada em conta a cada página navegada quando o assunto é *Internet Banking*. Agora, levando em consideração a abordagem do texto até aqui, levanta-se o questionamento: os aplicativos de *Internet Banking* também se preocupam com a privacidade do usuário quando este instala um aplicativo dessa categoria em seu celular?

Levando em consideração a quantidade de permissões utilizadas pelos aplicativos expostos até aqui, o aplicativo do banco Caixa mostra que sim, há a preocupação em manter privada a navegação do usuário enquanto este usufrui da ferramenta. Com apenas 18 permissões utilizadas, mesmo número informado pela PlayStore, o aplicativo avaliado é o que faz uso da menor quantidade de permissões entre os aplicativos de banco com mais downloads na PlayStore.

Para ilustrar, o Banco do Brasil informa que utiliza 20 permissões, contudo 23 são efetivamente utilizadas. O Banco Santander tem 27 permissões informadas e 25 permissões utilizadas. O Nubank tem 20 permissões informadas e 18 permissões efetivamente utilizadas.

5.2.5 WhatsApp Messenger

```
{"resultado": [{
      "permission": "read phone status and identity",
      "type": "Device ID & call information"
    }, {
      "permission": "retrieve running apps",
      "type": "Device & app history"
    }, {
      "permission": "directly call phone numbers",
      "type": "Phone"
    }, {
      "permission": "read call log",
      "type": "Phone"
    }, {
      "permission": "read phone status and identity",
      "type": "Phone"
    }, {
      "permission": "record audio",
      "type": "Microphone"
    }, {
      "permission": "precise location (GPS and network-based)",
      "type": "Location"
    }, {
      "permission": "approximate location (network-based)",
      "type": "Location"
```

```
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Storage"
    }, {
      "permission": "take pictures and videos",
      "type": "Camera"
    }, {
      "permission": "view Wi-Fi connections",
      "type": "Wi-Fi connection information"
    }, {
      "permission": "modify your contacts",
      "type": "Contacts"
    }, {
      "permission": "find accounts on the device",
      "type": "Contacts"
    }, {
      "permission": "read your contacts",
      "type": "Contacts"
    }, {
      "permission": "modify or delete the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "read the contents of your USB storage",
      "type": "Photos/Media/Files"
    }, {
      "permission": "read your own contact card",
      "type": "Identity"
    }, {
      "permission": "find accounts on the device",
      "type": "Identity"
    }, {
      "permission": "add or remove accounts",
      "type": "Identity"
    }, {
      "permission": "receive text messages (SMS)",
      "type": "SMS"
    }, {
      "permission": "send SMS messages",
      "type": "SMS"
    }, {
      "permission": "change your audio settings",
      "type": "Other"
    }, {
      "permission": "install shortcuts",
      "type": "Other"
    }, {
      "permission": "send sticky broadcast",
      "type": "Other"
    }, {
```

```
77        "permission": "use accounts on the device",
78        "type": "Other"
79      }, {
80        "permission": "toggle sync on and off",
81        "type": "Other"
82      }, {
83        "permission": "create accounts and set passwords",
84        "type": "Other"
85      }, {
86        "permission": "connect and disconnect from Wi-Fi",
87        "type": "Other"
88      }, {
89        "permission": "read sync settings",
90        "type": "Other"
91      }, {
92        "permission": "prevent device from sleeping",
93        "type": "Other"
94      }, {
95        "permission": "uninstall shortcuts",
96        "type": "Other"
97      }, {
98        "permission": "modify system settings",
99        "type": "Other"
100     }, {
101       "permission": "change network connectivity",
102       "type": "Other"
103     }, {
104       "permission": "run at startup",
105       "type": "Other"
106     }, {
107       "permission": "full network access",
108       "type": "Other"
109     }, {
110       "permission": "view network connections",
111       "type": "Other"
112     }, {
113       "permission": "read Google service configuration",
114       "type": "Other"
115     }, {
116       "permission": "pair with Bluetooth devices",
117       "type": "Other"
118     }, {
119       "permission": "control Near Field Communication",
120       "type": "Other"
121     }, {
122       "permission": "control vibration",
123       "type": "Other"
124     }]}
```

Trecho de código 5.11: Solicitação de permissões listadas listadas na PlayStore para o aplicativo Whatsapp Messenger, capturadas pelo módulo Scraper

```
1
2  Group: Location
3  Permissions: { "ACCESS_COARSE_LOCATION","ACCESS_FINE_LOCATION" }
4  Group: Camera
5  Permissions: { "CAMERA" }
6  Group: Microphone
7  Permissions: { "RECORD_AUDIO" }
8  Group: Calendar
9  Permissions: {}
10 Group: Phone
11 Permissions: { "READ_PHONE_STATE","MANAGE_OWN_CALLS" }
12 Group: Contacts
13 Permissions: { "READ_CONTACTS","WRITE_CONTACTS" }
14 Group: Storage
15 Permissions: { "WRITE_EXTERNAL_STORAGE","READ_EXTERNAL_STORAGE" }
16 Group: Sensors
17 Permissions: { "USE_FINGERPRINT","USE_BIOMETRIC" }
18 Group: SMS
19 Permissions: { "RECEIVE_SMS","SEND_SMS" }
20 Group: Call log
21 Permissions: {}
22 Group: Outros
23 Permissions: { "VIBRATE","AUTHENTICATE_ACCOUNTS","BROADCAST_STICKY","
       CHANGE_NETWORK_STATE","GET_ACCOUNTS","GET_TASKS","INSTALL_SHORTCUT","
       MANAGE_ACCOUNTS","MODIFY_AUDIO_SETTINGS","READ_PROFILE","READ_SYNC_SETTINGS","
       READ_SYNC_STATS","RECEIVE_BOOT_COMPLETED","USE_CREDENTIALS","WAKE_LOCK","
       WRITE_SYNC_SETTINGS","REQUEST_INSTALL_PACKAGES","FOREGROUND_SERVICE","
       USE_FULL_SCREEN_INTENT","INSTALL_SHORTCUT","UNINSTALL_SHORTCUT","RECEIVE","
       READ_GSERVICES","READ","WRITE","READ_SETTINGS","UPDATE_SHORTCUT","
       BROADCAST_BADGE","PROVIDER_INSERT_BADGE","READ_SETTINGS","WRITE_SETTINGS","
       CHANGE_BADGE","BROADCAST","MAPS_RECEIVE","REGISTRATION","READ","
       UPDATE_SHORTCUT","BROADCAST_BADGE","PROVIDER_INSERT_BADGE","READ_SETTINGS","
       WRITE_SETTINGS","CHANGE_BADGE"  }
24 Group: Network
25 Permissions: { "ACCESS_NETWORK_STATE","ACCESS_WIFI_STATE","BLUETOOTH","
       CHANGE_WIFI_STATE","INTERNET","NFC" }
```

Trecho de código 5.12: Permissões solicitadas pelo aplicativo Whatsapp Messenger em seu AndroidManifest, avaliado pelo módulo Analisador, e devidamente alocados em seus determinados grupos.

O Whatsapp é certamente um dos aplicativos de troca de mensagens mais populares do mundo. Oferece ao usuário desde suporte a mensagens de voz, envio e recebimento de arquivos de vários formatos, ferramenta de compartilhamento de pequenos vídeos e fotos para a lista de contatos do usuário, exibição de cartilhas em formato de *menu* comercial, entre outras ferramentas. (TEAM, 2019).

A figura abaixo (Figura 5.1), retirada do blog do aplicativo (WHATSAPP, 2019b), ilustra o caminho trilhado pelo mesmo ao longo do tempo e os incrementos em suas funcionalidades, sendo que todas esses incrementos refletiram em novas permissões pelo fato de fazerem uso de novos

recursos de sistema operacional e de hardware.

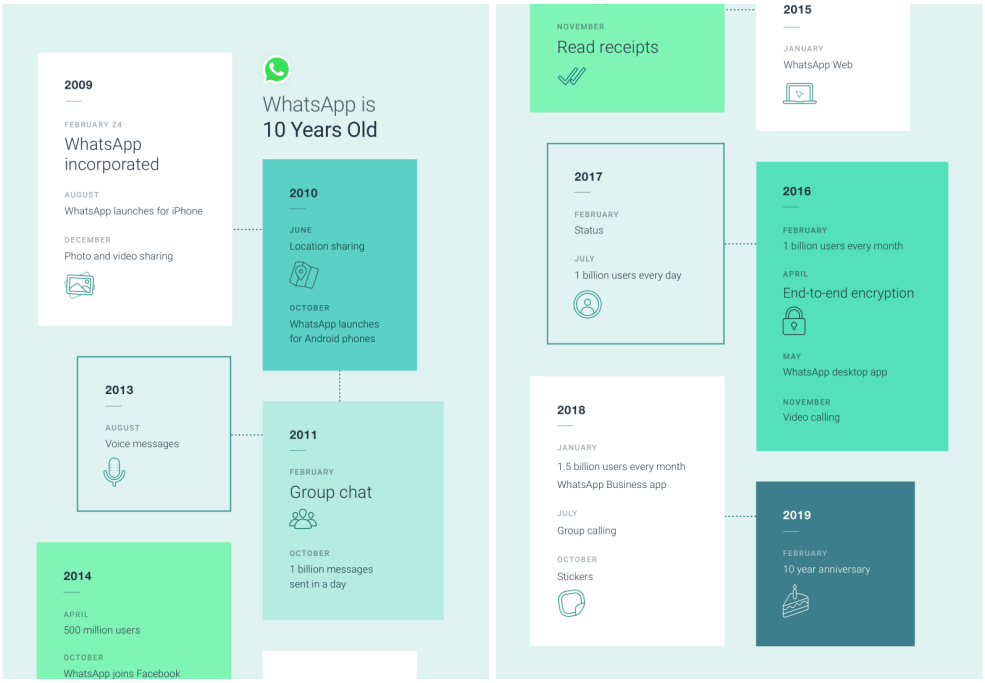

Figura 5.1: História do aplicativo baseado em suas implementações de maior relevância. Fonte: Blog de desenvolvimento Whatsapp

Em troca, o aplicativo também é campeão no número de permissões utilizadas, com a gritante quantia de 56 permissões utilizadas (quantidade avaliada pelo módulo Analisador), sendo que destas, 41 solicitações de permissões ao sistema operacionais estão listadas na PlayStore.

Em comparação, o concorrente Telegram conta com 53 permissões utilizadas e apenas a 33 informadas ao usuário.

O número elevado de permissões efetivamente utilizadas, assim como os aplicativos anteriores, deve-se a permissões de serviços inerentes às regras de negócio do aplicativo. As permissões BROADCAST, MAPS_RECEIVE, REGISTRATION e READ são do domínio de permissões *com.whatsapp.permission*, e é incerto afirmar sobre todos os recursos utilizados e implementados por essas permissões, pois não há documentação válida no site do desenvolvedor (WHATSAPP, 2019a) ou na documentação do Android que esclareça qual a finalidade de tais permissões.

Por fim, visto alguns casos de uso que validam a proposta deste trabalho e oferecem base para averiguar quantitativamente quantas solicitações de permissão são informadas ao usuário e quantas são realmente utilizadas, nas tabelas abaixo (tabelas 5.1, 5.2, 5.3 e 5.4) tais métricas.

Tabela 5.1: Grupo de aplicativos de natureza bancária.

Aplicativo	***Permissões informadas***	***Permissões utilizadas***
Banco do Brasil	20	23
Banco de Brasília	18	20
Bradesco	28	30
Caixa	18	18
Santander	27	25
Nubank	20	18

Tabela 5.2: Grupo de aplicativos de redes sociais e troca de mensagens instantâneas.

Aplicativo	***Permissões informadas***	***Permissões utilizadas***
Facebook	47	54
Twitter	30	40
Instagram	30	28
Whatsapp	41	56
Telegram	33	43

Tabela 5.3: Grupo de aplicativos de envio e recebimento de e-mails.

Aplicativo	***Permissões informadas***	***Permissões utilizadas***
GMail	32	42
Outlook	24	46
Yahoo! Mail	22	22

Tabela 5.4: Aplicativos diversos.

Aplicativo	***Permissões informadas***	***Permissões utilizadas***
Spotify	25	28
Uber	34	30
Mercado Livre	19	37
OLX	15	29

Capítulo 6

Conclusão

A ideia original deste projeto era a exibir para o usuário do dispositivo mobile as permissões que os aplicativos instalados utilizavam, sem o devido conhecimento deste. Porém, a proposta evoluiu de tal forma que apenas o estudo sobre `AndroidManifest` existente no sistema operacional Android desencadeou uma sequência de questionamentos acerca da metologia utilizada pelos desenvolvedores do sistema operacional no que tange as permissões de serviços concedidos a aplicativos terceiros.

Num cenário onde os dispositivos móveis fazem parte do dia a dia como evidenciado no capítulo introdutório deste projeto, levando em conta que vazamentos informações pessoais são recorrentes e vem de empresas dos mais variados portes, a preocupação com a segurança de informações privadas dos usuários destes dispositivos deve ser motivação para o desenvolvimento de ferramentas que busquem maior transparência em relação aos riscos envolvido com o uso dos aplicativos, o que vai muito além de simplesmente "exibir uma janela de notificação"informando que o usuário está autorizando o consumo de algumas informações.

Para tanto, a avaliação de como o arquivo `AndroidManifest` é montado levanta questionamentos sobre permissões, grupos de permissões, níveis de proteção de cada uma delas e abre espaço para discussão sobre melhorias no sistema operacional quanto a esse ponto, e obviamente é importante ter como principal preocupação a segurança da informação do usuário.

A ferramenta desenvolvida atende aos requisitos e concepção informadas no capítulo 3, quando estabelece-se que a abordagem utilizada será de análise, validação e contestação de permissões utilizadas por aplicativos, almeja-se, como exposto na análise de resultados, inspirar o usuário a ponderar sobre quais de suas informações pessoais tem integridade comprometida ao utilizar os aplicativos instalados em seu celular, e auxilia o mesmo a concluir que, por mais despretensiosa que seja sua utilização de aplicativos, seus dados estão a todo instante sendo monitorados por meio de acesso a permissões questionáveis que usam máscaras de uma categoria, e em certos casos espionam e vendem informações pessoais.

Ainda sobre a ferramenta, é importante evidenciar que sua intenção é meramente expositiva no que tange as permissões utilizadas por aplicativos mobile, e que há espaço para melhora e acréscimo de seus módulos, como a implementação de exibição a partir do nível de segurança que esta tem

no `AndroidManifest.xml` do sistema operacional. Outras melhorias cabem também na avaliação entre pares ou grupos de aplicativos de dada categoria em relação a um selecionado, evidenciando permissões utilizadas similares entre estes e destacando aquelas solicitações de permissão que são de caráter duvidoso quando levado em conta a categoria do aplicativo, como exemplificado pela análise do aplicativo OLX.

6.1 Trabalhos Futuros

Tendo como base o que foi estudado e desenvolvido durante este projeto é possível notar que há necessidade do desenvolvimento de ferramentas que permitam maior transparência em relação às permissões de serviço que são utilizadas por aplicativos Android, assim como a quais informações essas permissões permitem acesso.

Algumas possibilidades interessantes para suprir as necessidades apresentadas é o desenvolvimento de bibliotecas embarcadas no sistema operacional. Essas bibliotecas teriam objetivo de gerar continuamente logs para manter o usuário informado a respeito de quais informações estão sendo transitadas entre os sistema operacional e seus aplicativos. Outra opção seria desenvolvimento de aplicativos mobile semelhantes à plataforma web apresentada neste projeto, mas com o mesmo fim: investigar permissões de serviço em uso por aplicativos em dispositivos Android.

No mais, concluímos que a ferramenta desenvolvida aborda os requisitos levantados pela proposta estabelecida e possui relevância no contexto atual, onde a maioria da população está conectada à Internet e consome aplicativos para Android por meio de seus dispositivos móveis e, dada a pouca instrução oferecida sobre como consumir serviços se prevenindo de riscos de exposição são vítimas de crimes cibernéticos.

Bibliografia

ALTHOFF, Tim; WHITE, Ryen W; HORVITZ, Eric. Influence of Pokémon Go on Physical Activity: Study and Implications. **J Med Internet Res**, v. 18, n. 12, e315, dez. 2016. ISSN 1438-8871. DOI: 10.2196/jmir.6759. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/27923778>.

CALADO, Ricardo. **Features**. 2019. Disponível em: <https://blog.geekhunter.com.br/angular-8-novidades-da-versao/>. Acesso em: 25 nov. 2019.

CASTRO, Pablo Samuel et al. From taxi GPS traces to social and community dynamics: A survey. **ACM Comput. Surv.**, v. 46, 2013.

CISCO. **Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper**. 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>. Acesso em: 22 jun. 2019.

CTHACKERS. **ADM-ZIP for NodeJS with added support for electron original-fs**. 2018. Disponível em: <https://www.npmjs.com/package/adm-zip>. Acesso em: 10 nov. 2019.

DEVELOPERS, Google. **Android Runtime (ART) and Dalvik**. 2019. Disponível em: <https://source.android.com/devices/tech/dalvik/index.html>. Acesso em: 13 maio 2019.

_____. **API Packages**. 2019. Disponível em: <https://developer.android.com/reference/packages.html>. Acesso em: 12 maio 2019.

_____. **App Manifest Overview**. 2019. Disponível em: <https://developer.android.com/guide/topics/manifest/manifest-intro>.

_____. **App Manifest Permissions**. 2019. Disponível em: <https://developer.android.com/reference/android/Manifest.permission>.

_____. **Application Sandbox**. 2019. Disponível em: <https://source.android.com/security/app-sandbox>. Acesso em: 10 nov. 2019.

_____. **Content Providers**. 2018. Disponível em: <https://developer.android.com/guide/topics/providers/content-providers>. Acesso em: 1 jul. 2019.

_____._____. 2019. Disponível em: <https://developer.android.com/things/sdk/drivers/location>. Acesso em: 14 maio 2019.

_____. **HAL Types**. 2019. Disponível em: <https://source.android.com/devices/architecture/hal-types>. Acesso em: 17 maio 2019.

DEVELOPERS, Google. **Permissions overview**. 2019. Disponível em: <https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous>. Acesso em: 10 nov. 2019.

_____. **Platform Architecture**. 2019. Disponível em: <https://developer.android.com/guide/platform>. Acesso em: 30 jun. 2019.

_____. **Protection Level**. 2019. Disponível em: <https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous>. Acesso em: 10 nov. 2019.

_____. **Sign your app**. 2019. Disponível em: <https://developer.android.com/studio/publish/app-signing>. Acesso em: 11 nov. 2019.

_____. **System and kernel security**. 2019. Disponível em: <https://source.android.com/security/overview/kernel-security.html>. Acesso em: 22 jun. 2019.

_____. **Tools Overview**. 2009. Disponível em: <https://developer.android.com/studio/command-line>. Acesso em: 27 abr. 2019.

DURAN, Paulina. **Two Australian banks among six targeted by fake apps: security firm**. 2018. Disponível em: <https://www.reuters.com/article/us-australia-technology-fraud/two-australian-banks-among-six-targeted-by-fake-apps-security-firm-idUSKCN1M00FE>. Acesso em: 20 jun. 2019.

GABBIADINI, Alessandro; SAGIOGLOU, Christina; GREITEMEYER, Tobias. Does Pokemon Go lead to a more physically active life style? **Computers in Human Behavior**, v. 84, p. 258–263, 2018. ISSN 0747-5632. DOI: https://doi.org/10.1016/j.chb.2018.03.005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0747563218301109>.

GUARDIAN. **Open ports to a VM using the Azure portal**. 2018. Disponível em: <https://guardianapp.com/research/ios-app-location-report-sep2018/>. Acesso em: 16 jun. 2019.

GUPTA, Aditya. **Hacking your Droid**. 2019. Disponível em: <https://dl.packetstormsecurity.net/papers/presentations/HackingyourDroid-Slides.pdf>. Acesso em: 11 nov. 2019.

HENRIQUE GONCALVES SA, Joao et al. GeoHealth: A Georeferenced System for Health Data Analysis in Primary Care. **Latin America Transactions, IEEE (Revista IEEE America Latina)**, v. 10, p. 1352–1356, jan. 2012. DOI: 10.1109/TLA.2012.6142483.

KAMEL BOULOS, Maged; YANG, Stephen. Exergames for health and fitness: The roles of GPS and geosocial apps. **International journal of health geographics**, v. 12, p. 18, abr. 2013. DOI: 10.1186/1476-072X-12-18.

KOTTASOVÁ, Ivana. **It won't even notice.** 2018. Disponível em: <https://edition.cnn.com/2018/10/25/tech/facebook-fine-data-scandal/index.html>. Acesso em: 21 jun. 2019.

KOVACS, Eduard. **Android Manifest File Attacks Can Make Devices Inoperable**. 2019. Disponível em: <https://www.securityweek.com/android-manifest-file-attacks-can-make-devices-inoperable>. Acesso em: 8 jan. 2015.

LABFORENSE. **Sistema Móvel de Suporte à Perícia - AppINC**. 2019. Disponível em: <https://github.com/paulohlips/appINC/tree/master/Documenta%5C%C3%A7%5C%C3%A3o>. Acesso em: 24 nov. 2019.

LEARNINGOS. **Operating Systems Definition and the Classification of OS**. 2018. Disponível em: <http://learningoperatingsystem.blogspot.com/2015/09/operating-systems-definition-and.html>. Acesso em: 21 jun. 2019.

MEDIUM. **Get smarter about what matters to you.** 2019. Disponível em: <https://medium.com>. Acesso em: 11 nov. 2019.

NODEJS. **Node.js v10.17.0 Documentation**. 2018. Disponível em: <https://nodejs.org/docs/latest-v10.x/api/>. Acesso em: 10 nov. 2019.

NPM. **Application Sandbox**. 2018. Disponível em: <https://medium.com/npm-inc/this-year-in-javascript-2018-in-review-and-npms-predictions-for-2019-3a3d7e5298ef>. Acesso em: 11 nov. 2019.

PHANDROID.COM. **Android PSA: Stop Using Task Killer Apps**. 2011. Disponível em: <https://phandroid.com/2011/06/16/android-psa-stop-using-task-killer-apps-now/>. Acesso em: 29 jun. 2019.

PRUDENTE, Neemias. **Monitoramento eletrônico: uma efetiva alternativa a prisão?** [S.l.: s.n.], 2014. Disponível em: <https://neemiasprudente.jusbrasil.com.br/artigos/121942848/monitoramento-eletronico-uma-efetiva-alternativa-a-prisao>.

ROSE, Geoff; LTD Francis and, Sgm. Mobile phones as traffic probes: practices, prospects and issues. **Transport Reviews**, v. 26, p. 275–291, abr. 2006.

ROTOLO, Paolo. **How Libraries can silently add permissions to your Android App**. 2019. Disponível em: <https://medium.com/glucosio-project/how-libraries-can-silently-add-permissions-to-your-android-app-620911d7de6c>. Acesso em: 15 mar. 2016.

SILVA, Matheus Rabelo da. **Monitoramento eletrônico de presos: Tornozeleira Eletrônica**. [S.l.: s.n.], 2016.

STATCOUNTER. **Mobile Tablet Android Version Market Share Worldwide - June 2019**. 2019. Disponível em: <http://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>. Acesso em: 29 jun. 2019.

TAQTILE. **Híbrido x Nativo**. 2016. Disponível em: <https://medium.com/taqtilebr/h%5C%C3%5C%ADbrido-vs-nativo-c8591df0dce6>. Acesso em: 10 nov. 2019.

TEAM, Whatsapp Development. **Whatsapp Development Blog**. 2019. Disponível em: <https://blog.whatsapp.com/>. Acesso em: 10 nov. 2019.

TRUSTLOOK. **How to Stop Mobile Apps That Steal**. 2018. Disponível em: <https://securityintelligence.com/how-to-stop-mobile-apps-that-steal/>. Acesso em: 23 jun. 2019.

WHATSAPP. **FAQ**. 2019. Disponível em: <https://faq.whatsapp.com/pt_br/android/>. Acesso em: 25 nov. 2019.

_____. **Features**. 2019. Disponível em: <https://blog.whatsapp.com/?lang=pt_br>. Acesso em: 25 nov. 2019.

ZAIATZ, Matheus Felipe. **Análise da mobilidade dos usuários do Campus Darcy Ribeiro da Universidade de Brasília utilizando dados de telefonia móvel**. [S.l.: s.n.], 2018.